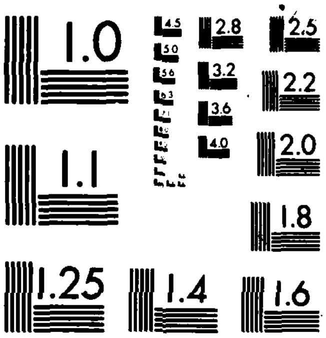
AD-A141 539 PROGRAM PREGA PIVOTAL REGRESSION ANALYSIS OF GAMMA-RAY 1/3
SPECTRA FROM NAI(T.. (U) NAVAL RESEARCH LAB WASHINGTON
DC G W PHILLIPS ET AL. 06 APR 84 NRL-MR-5275

UNCLASSIFIED SBI-AD-E000 564 F/G 9/2 NL



1.0
4.5
5.0
5.6
6.3
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

2

ADE000 564

NRL Memorandum Report 5275

[illegible]

06 21 042

AD-A141 539

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
NRL Memorandum Report 5275

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Naval Research Laboratory

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State and ZIP Code)
Washington, DC 20375

7b. ADDRESS (City, State and ZIP Code)

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Office of Naval Research

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State and ZIP Code)
Arlington, VA 22217

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
31327N	R0113		66-0403-A3

11. TITLE (Include Security Classification)
(See Page ii)

12. PERSONAL AUTHOR(S)
G.W. Phillips and B.G. Glagola

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM TO

14. DATE OF REPORT (Yr., Mo., Day)
April 6, 1984

15. PAGE COUNT
192

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
NaI detectors
Gamma-ray spectra
Regression analysis
PREGA

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Program PREGA was developed at NRL to perform a fast and efficient regression analysis of a gamma-ray spectrum from an unknown source, taken with a NaI(Tl) detector. The analysis determines the best subset from a library of standard spectra to use in fitting the data. Output at each step includes relative intensities and uncertainties for each standard used in the fit, plus the partial F statistic for significance of each library element. Final output includes an analysis of variance (ANOVA) table, distribution of residuals, location of largest residuals, distribution of runs of consecutive residuals with the same sign, and the location of largest runs. The program was written for the Nuclear Data ND6620 computer and has been transferred to VAX II/780.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. □ DTIC USERS □

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
G. W. Phillips

22b. TELEPHONE NUMBER (Include Area Code)
(202) 767-5692

22c. OFFICE SYMBOL
Code 6608

11. TITLE (Include Security Classification)

PROGRAM PREGA — Pivotal Regression Analysis of Gamma-Ray Spectra from NaI(Tl) Detectors for the ND6620 Computer

CONTENTS

Program PREGA
Pivotal Regression Analysis of Gamma-Ray Spectra from NaI(Tl) Detectors for the ND6620 Computer

1. INTRODUCTION

Gamma-ray pulse-height spectra from NaI(Tl) detectors often contain few or no discernable peaks due to the source of interest, as a result of the poor intrinsic resolution of these detectors and a predominance of background gamma rays. Even after background subtraction, the resulting spectrum is often of too poor a quality to reveal characteristic peaks. Thus, one cannot do a peak analysis such as is usually performed on gamma-ray spectra from high-resolution germanium diode detectors (see Ref. 1).

However, there is often sufficient information contained in such spectra in the overall spectral shape to identify the source of the gamma rays by comparison to a library of standard source shapes. The situation becomes more complicated when the spectrum may be due to a superposition of gamma-ray spectra from two or more sources. To determine the relative contributions of the individual sources it is necessary to do a multiple least-squares fit of the unknown source spectrum to a library of standard source spectra using regression analysis techniques.

The ALPHA-M program was developed at Oak Ridge[2] to perform such a fit and calculate the concentrations of each of the standards in the unknown.

However, if there are more than a very few standards in the library, such a procedure often gives an ambiguous result, due to correlations between library spectra. In such a case, it is necessary to do a systematic search for the subset of library standards which gives the best fit to the unknown spectrum. Program PREGA was developed at NRL to efficiently perform this search for the best subset. It works in one of two optional modes.

The first mode is similar to standard stepwise regression analysis methods. It begins with a subset containing only background, and performs a least-squares fit for the intensity of the background in the unknown spectrum. It then calculates the correlation of the residual spectrum, not accounted for by background, with each of the remaining standards in the library. That standard with the largest correlation is added to the fitting subset and a new least-squares fit is performed. This continues until no remaining standards are significantly correlated with the residuals.

Manuscript approved November 29, 1983.

The second mode begins with a fit to the full library. From the results of this fit a test set of library members most correlated with the unknown spectrum is selected. Additional library members are then pivoted in and out of the test set, depending on their correlation with the residuals, until the subset which gives the best significant fit is obtained.

PREGA begins by calculating the matrix of cross-correlations of the library members with each other and with the unknown spectrum. It then operates on this matrix using a method by Garside[3] to obtain the quantities of interest in the fit with a minimum of mathematical manipulations. At no time is it necessary to do a complete matrix inversion, and individual library members can be pivoted into and out of the fit quickly and simply. This method is thus tailor-made for either the stepwise mode or the full library mode of PREGA.

The program PREGA is implemented as an extensive modification of the program NAI supplied by Nuclear Data, Inc.[4] and runs on the Nuclear Data ND 6620 system. It is written in DEC RT-11 FORTRAN and is currently being adapted to run on the Digital Equipment Corporation VAX 11/780.

2. SETUP FOR ANALYSIS

Preparation of Library

Library spectra should be taken for each source of interest under conditions as close as possible as those under which the unknown spectrum is taken. Backscattering of gamma rays by nearby materials can affect the shape of the spectrum as can attenuation due to intervening materials between the source and detector. This results in removal of counts from the full-energy peaks and the addition of counts with lower energy due to scattered gamma rays.

Background spectra taken before and after each library spectrum should be normalized by counting time and subtracted from the library spectrum. An energy calibration must be determined either from known peaks in the library or background spectra, or from a calibration source spectrum taken before and after each library spectrum. Program PKANAL is included in the PREGA package and may be used to determine the position of the peaks. At least two peaks are needed, well separated in energy, to determine a linear calibration of peak energy E versus channel number I in the form

$$E(I) = E0 + I*E1.$$

If more than two peaks are available a linear least squares fit can be used to determine the offset E0 and slope E1. These should be recorded with each library spectrum along with source type, configuration, strength, range, attenuation and any other pertinent information.

Preparation of Background

A background spectrum should be taken as close as possible in time and other conditions to the unknown, preferably immediately before and after. An energy calibration should be obtained in the same manner as for the library spectra. Program GSHIFT can then be used to shift the slope and offset of each library member to match the slope and offset of the unknown spectrum. PREGA has the option of subtracting the background from the unknown before fitting. Nevertheless, the background spectrum should be included in the library, because background intensity changes could lead to over or under subtraction. If this happens, a negative intensity may result for the background. To allow for this PREGA has the option of not allowing the background to be pivoted out. (A negative intensity for any other library member is generally not physically allowable and should be rejected).

Preparation for PREGA

The library to be used in the fit is specified interactively in the program SEDIT. The standard name, file name, counting time and half-life activity are specified for each library member. A background spectrum, which may be subtracted from each library member, must be specified although this is generally not used.

The unknown and its background are specified interactively in the program UEDIT. This also allows the user to select various options to be used in the fit and to select which of the library members will be used. The information entered by the user in SEDIT and UEDIT are written to control files which are used by PREGA to perform the fit. Thus PREGA requires no direct input from the user and can be run as a batch job if desired. Output is usually to a file which can be listed later on the lineprinter.

Selection of Mode

If only one or two library members are expected to be present together in the unknown, the stepwise mode of PREGA is usually more efficient than the mode which begins by using the entire library in the fit. If multiple library members may be present, it is usually best to run both modes. They will generally, but not always, agree on the subset of library members which give the best fit. In cases where they do not agree, correlations between library members or linear combinations of library members will be found to cause ambiguities in the fit. Enough information is printed out at each step for the analyst to determine which library members are causing the ambiguity. They can either be eliminated from the library or accepted as doubtful contributions to the unknown spectrum.

3. ALGORITHMS

Partial F Test

The criterion used for selection or rejection of library members at each step is given by the partial F-test statistic F_p defined in Chapter 4. For convenience, this is converted to an equivalent normally distributed parameter X(F) which gives the value of F_p in standard deviations (sigma units). This is compared to a threshold F0 usually set at 2.5 sigma.

Full PREGA

The full PREGA algorithm begins with the fit to the complete library ana selection of the trial subset C. It then goes directly to the regression on the trial subset C. PREGA then pivots library elements out (backward step) and in (forward step) successively, based on the partial F-values, until a best fit is found. The full PREGA algorithm proceeds as follows:

```
1. Do the fit to the complete library L.

2. Let the trial subset C include all library elements I with a
   relative intensity $B_I$ and error $\sigma_I(B)$ such that

        $B_I/\sigma_I(B) > 1.$

3. Let E be the set containing all library elements included in
   the fit at each step. Initially let $E = C$.

4. Let D be the set of elements not in the fit at the beginning
   of step 6, below. Initially let $D = 0$ (empty set).

5. Backward Pivot: to check for removal of library elements
   from subset E.

   a. Let $C = E$.

   b. Calculate partial F to remove, $F_p(I)$, for all library
      elements I in C.

   c. Let $F_p(J) = \min F_p(I)$, for all I in C, and convert
      to $X(F)_J$ in sigma units.

   d. IF: $X(F)_J < F0$,
          THEN: remove J, let $E = E - J$;
          ELSE: continue.

   e. IF: $E = L-D$, (no change in trial subset)
          THEN: end regression (go to 7);
          ELSE: continue.

6. Forward Pivot: to check for addition of library elements not
   in C during step 5.

   a. Let $D = L-C$

   b. Calculate partial F to include, $F_p(I)$, for all library
      elements I in D.

   c. Let $F_p(J) = \max F_p(I)$, for all I in D, and convert
      to $X(F)_J$ in sigma units.
```

```
d.  IF:  $X(F)_J > FO$,
         THEN:  include J, let $E = E+J$;
         ELSE:  continue.

e.  IF:  $E = C$, (no change in trial subset)
         THEN:  end regression (go to 7);
         ELSE:  repeat pivots, go to 5.
```

7. End regression: do analysis of variance, calculate residuals and runs, and print out results.

Stepwise PREGA

The stepwise PREGA mode is implemented as an option of program PREGA. It is run by setting the Background Regression (BR) parameter in UEDIT to 1: trial set = background only. The stepwise mode proceeds as follows:

1. Begin regression with fit to background only.

2. Let the trial subset C = (background).

3. Initially let E = C.

4. Go to the forward pivot (step 6) of the full PREGA mode and proceed from there on as in the full PREGA.

4. FORMULAS AND DEFINITIONS

1. Let Y be the vector of the observed source data spectrum,

$$Y = \{y(1), y(2), \ldots, y(N)\},$$

and let X_I be a similar vector for the spectrum of library element I.

2. Define the augmented cross-product matrix S_L of dimension K, where the number of library elements is K-1, by

$$S_L = \left\{ S_{IJ} \right\} \quad \text{for } I,\ J = 1,\ \ldots,\ K$$

where

$$\begin{aligned} S_{IJ} &= (X_I \,.\, S_J), \text{ for } I,\ J=1,\ \ldots,\ K-1 \\ S_{IK} &= (X_I \,.\, Y) \\ S_{KJ} &= (Y \,.\, X_J) \\ S_{KK} &= (Y \,.\, Y), \end{aligned}$$

and the dot product

$$(X_I \, . \, X_J) \triangleq \sum_{r=1}^{N} w(r) \; x_I(r) \; x_J(r)$$

The weights w(r) are calculated by the inverse Poisson variances of the data y(r).

3. The least squares solution for

$$Y = B \, X + \epsilon,$$

where ϵ is the vector of random errors in the data, is given by

$$B = (X \cdot X)^{-1}(X \cdot Y)$$

where B is the vector of coefficients of the library elements in the fit,

$$B = \{B_1, \ldots, B_{k-1}\}.$$

4. The residual sum of squares is given by

$$\begin{aligned} RSS &= (Y - BX)^T \cdot (Y - BX) \\ &= (Y \cdot Y) - (X \cdot Y)^T (X \cdot X)^{-1} (X \cdot Y). \end{aligned}$$

5. The results of a regression of Y on a subset C of the library L can be completely determined from the matrix S_C* obtained by successive pivots ϕ on the matrix S_L, for each library element in C. The pivot operation used in PREGA is described by Garside.[3] The results for N data points and P elements I in C are

$$\begin{aligned} B_I &= S_{IK}{}^* \\ RSS_C &= S_{KK}{}^* \\ \hat{\sigma}_C{}^2 &= RSS_C/(N-P) \\ \sigma_I(B)^2 &= S_{II}{}^* \, \hat{\sigma}_C{}^2 \end{aligned}$$

6. The partial F-test[4] to remove library element I from C is given, for H = C-I, by

$$F_p^-(I) = (RSS_H - RSS_C)/(RSS_C/(N-P))$$

$$= B_I^2/\sigma_I(B)^2$$

$$= (N-P)(S_{IK}^*)^2/(S_{II}^* S_{KK}^*)$$

then $$RSS_H = RSS_C\ (1 + F_p^-/(N-P)).$$

7. The partial F test to include library element J not in C is given, for G = C+J, by

$$F_p^+(J) = (RSS_C - RSS_G)/(RSS_G/(N-P-1))$$

where

$$RSS_G = RSS_C(1 - R_{JK.C}^2)$$

and the partial correlation coefficient of J and K given C is just[6]

$$R_{JK.C}^2 = (S_{JK}^*)^2/(S_{JJ}^*\ S_{KK}^*).$$

8. An approximate normal transformation for F with D,1 degrees of freedom is used,*

$$X(F) \approx \sqrt{F}\ (1 - 1/4D)/(1 - F/2D)^{1/2}$$

(note that for D>>1 and D>>F then $X \approx \sqrt{F}$).

9. An approximation for the probability of obtaining a random value greater than X is used,*

$$Q(X) \approx Z(X)(a_1t + a_2t^2 + a_3t^3)$$

where $$Z(X) = \exp(-\ X^2/2)/(2\pi)^{1/2},$$

$$t = 1/(1 + px),$$

and $$a_1 = .436184,\ a_2 = -\ .120168,$$
$$a_3 = .937298,\ p = .33267.$$

*See the "Handbook of Mathematical Functions," edited by M. Abramowitz and I.A. Stegum, Dover Publications, New York (1965).

5. PROGRAM PKANAL

Operation

Program PKANAL reads in data from ND spectral files and calculates a background, peak positions, width and area for selected regions.

Language

The program is written in DEC RT11 FORTRAN and runs on the Nuclear Data ND6620 computer under the MIDAS operating system.

Inputs

Diskfile — logical unit (LU) 8, spectral data in ND format

Keyboard — LU 5, operator inputs first and last channels for up to 20 regions followed by one or more filenames containing spectra to be analyzed

Outputs — LU6, table of results for each spectrum

Input Variables

Record 1a,...
JL, JR — lower and upper limits of each region to be analyzed, one pair of limits per record up to a maximun of twenty (20), terminated with a double carriage return

Record 2a,... — filenames for spectra to be analyzed, one per record, terminated with a double carriage return

Lineprinter Output

Heading	Contents
REGION LIMITS	input limits (JL,JR) for region
PEAK LIMITS	limits determined by the program for the peak
HEIGHT	maximum peak height above background
AREA	net peak area above background
BKGD	background area beneath the peak
CENTROID	calculated peak position (center of gravity)
VARIANCE	calculated variance about centroid
FWHM	peak full width at half maximum
ENERGY	calibrated energy of centroid position

Subroutines Called

FREEFM	free field input routine
GET	reads spectrum from ND spectral files
PEAKNL	does peak analysis for each region

6. PROGRAM GSHIFT

Operation

Program GSHIFT reads in a spectrum and its energy calibration from disk, performs a specified gain shift and zero offset, and writes the shifted spectrum and its calibration back to disk.

Language

The program is written in DEC RT11 FORTRAN and runs on the Nuclear Data ND6620 computer under the MIDAS operating system.

Inputs

Diskfile	logical unit (LU) 8, spectral data in ND format
Keyboard	LU5, operator inputs filenames and selects new gain or desired shift

Outputs

Diskfile	LU8, shifted spectrum in ND format
CRT	LU5, prompts to operator, energy calibration from header, etc.

Input from Keyboard

Rec.	Prompt	Response
1.	FILENAME FOR SPECTRUM	FILE.ELEMENT (for input spectrum)
2a.	ENERGY CALIB. FROM HEADER GIVES . . . . USE THIS . . .	YES (go to 3a.), or NO (go to 2b), (default = NO)*
2b.	ENTER OLD SLOPE, OFFSET	slope, offset (for calibration of input spectrum)

3a.	TO GET NEW SLOPE AND OFFSET. . .	FILE.ELEMENT (spectrum with desired slope and offset in header, go to 4), (default, go to 3b)*
3b.	ENTER NEW SLOPE, OFFSET	slope, offset (for desired calibration, go to 4), (default, go to 3c)*
3c.	ENTER GAIN, ZERO SHIFT	desired gain, zero shift
4.	FILENAME FOR OUTPUT SPECTRUM	FILE.ELEMENT

*immediate carriage return gives default values

Subroutines Called

GET reads spectrum and energy calibration from ND spectral file

FREEFM free field input routine

REGAIN performs gain and zero shift

RECHAN restores proper channel offset after call to REGAIN

PUT writes spectrum and energy calibration to ND spectral file

7. PROGRAM PREGA

Operation

Program PREGA performs a stepwise or pivotal regression analysis in order to fit a data spectrum to the "best" subset of a library consisting of a background spectrum and several characteristic source spectra. The program is essentially a modification to the Nuclear Data Inc. ND6600 NAI Data Reduction Package,[5] designed to make analysis more efficient for the user. Therefore, the basic package architecture used by Nuclear Data Inc. is still retained as far as program input, file structure and logical units used is concerned. The general flow of PREGA is given in the flow chart in Fig. 1. A more detailed flow chart is given in Appendix B.

Language

The program is written in DEC RT11 FORTRAN and runs on the Nuclear Data ND6620 computer under the MIDAS operating system.

The program PREGA does not accept information directly from the user to perform its calculations; parameters and calculation options are entered through, and edited by, the editor programs SEDIT and UEDIT. All spectral data acquired by the user are stored on disk files and automatically read by PREGA; the user has only to indicate, via the editor programs, the locations of the spectral data.

For the purposes of this discussion, all input data may be grouped into six categories:

(1) spectral data
(2) individual library standard parameters
(3) all library standard parameters and calculation options
(4) sample spectrum parameters
(5) sample background parameters and calculation options
(6) weighting factor calculation options

On the following pages are listed all the necessary input data for the PREGA Program; where user options are available, they are mentioned.

For a more complete description of the input data, the user is referred to the section containing the listings and descriptions of the programs UEDIT and SEDIT. The actual use of UEDIT and SEDIT by PREGA is referred to in the section on software organization.

(1) Spectral Data

Gamma-ray spectra are acquired by the user in the laboratory and written on a disk file in spectral data format.

(2) Individual Library Standard Parameters

a. (standard number: 1 to 10 assigned by SEDIT in sequential order)

b. standard name: 8 characters maximum
c. location of individual standard spectrum on disk file LU-8: FILE.ELEMENT(,DEVICE)
d. half-life
e. half-life units: Y,D,H,M or S
f. counting time in seconds
g. activity (concentration): usually set to 1, or may be set to convenient units

(3) All Library Standard Parameters and Calculation Options

a. number of channels per standard: maximum 512
b. initial channel number
c. final channel number
d. location of background spectrum on disk file LU-8: FILE.ELEMENT(,DEVICE)
e. counting time for background spectrum, in seconds
f. background subtracted from spectrum: Yes or No ?
g. number of isotopes used from library
h. printout order of library standards
i. library standard rejection coefficient; usually $B/\sigma < 1$
j. rejection coefficient applied Yes or No ?
k. F-test rejection coefficient; usually set to 2.5

(4) Sample Spectrum Parameters

a. location of sample spectrum on disk file LU-8: FILE.ELEMENT(,DEVICE)
b. counting time, in seconds
c. volume reduction factor; usually set = 1
d. decay time, in seconds; set = 1
e. result multiplication factor; usually set = 1

(5) Sample Background Parameters and Calculation Options

a. location of sample background spectrum on disk file LU-8: FILE.ELEMENT(,DEVICE)
b. background supplied: Yes or No
c. is background to be subtracted from the sample spectrum Yes or No
d. counting time of background spectrum, in seconds

(6) Weighting Factor Calculation Options

a. How determined; user has 2 options:
option 1: based on original spectral counts and held constant
option 2: based on least-squares spectrum counts
b. weighting factor calculation: user has 3 options (p_i = weighting factor, y_i = sample counts, b_i = normalized bkg counts)
option 1: $p_i = 1/y_i$
option 2: $p_i = 1/(y_i+b_i)$
option 3: $p_i = 1$

Outputs

The output device, usually a line printer, prints out the following items in the order indicated: (see examples in Appendix A).

(1) sum of standards
(2) sample spectrum counts in each channel
(3) sum of background and sample counts
*(4) "BACKGROUND NOT INCLUDED AS A STANDARD"
*(5) "PIVOT ON BACKGROUND PERMITTED"
(6) correlation matrix
*(7) "STARTING ANALYSIS WITH TRIAL SET=BACKGROUND ONLY"
(8) Result of current regression
(9) Final Result
(10) LLNL R-STATISTIC
(11) Final Statistics for eliminated standards
(12) residuals/standard deviations per channel
(13) suspicious channels
(14) Distribution of residuals
(15) Distribution of Runs
(16) locations and sizes of large runs

*These items are intended as visual flags in the output for the analyst.

(1) "SUM OF STANDARDS"

For each standard nuclide in the complete library, the program gives the total number or counts in all channels recorded during the counting time of the radionuclide. These sums are printed out in the same order as the nuclides appear in the library. For the example shown in Appendix A, the complete library contained five (5) standards. In fitting the sample spectrum only four (4) of these standards are actually used. This is the last stage of the program in which the complete library is handled. All subsequent calculations only use the library standards (in the order in which they are listed in variable OR) given in the program UEDIT.

(2) "SAMPLE SPECTRUM"

On the first line is printed the disk file location (LU-8) of the sample spectrum FILE.ELEMENT(,DEVICE). The sample counts minus the background counts for each channel are then printed. The background counts are multiplied by the ratio of the counting time of the sample counting time of the background to correct for any differences in counting times. The printout should be read horizontally from left to right.

(3) "BACKGD SUM" and "SAMPLE SUM"

These are the background and sample counts in each channel (see "SAMPLE SPECTRUM" listed above) summed from the initial channel for computation to the final channel for computation. See page 36, program lines 129 and 134.

(4) "BACKGROUND NOT INCLUDED AS A STANDARD"

This is an output flag that is printed only when the background is not included as one of the selected library standards. **NOTE** For this flag to be functional the background spectrum must be standard No. 1 in the complete library list. The user is referred to the section on the program SEDIT.

(5) "PIVOT ON BACKGROUND PERMITTED"

This flag is output when the variable BP in UEDIT is set equal to one (BP = 1). It means that the user wants to allow the program to perform pivot operations on the background if the calculation allows them. **NOTE** This flag requires the background spectrum to be standard No. 1 in the full library (see SEDIT).

(6) "CORRELATION MATRIX"

Tnis matrix is calculated to show the correlation of the user chosen set of standards with themselves and the sample. The standards are output in tne order given by the variable OR in UEDIT. The sample always occupies the last row and column of the matrix.

(7) "STARTING ANALYSIS WITH TRIAL SET = BACKGROUND ONLY"

This flag is output when the user wishes to begin the regression with the trial set chosen to be only the background spectrum (UEDIT variable BR = 1). In this case PREGA executes in a manner similar to a step-wise regression (STREGA). **NOTE** This option flag requires that the background spectrum is standard No. 1 in the full library set (see SEDIT).

(8) Results of current regression

This section of output contains several pieces of information. These are:

a) A title line stating the type of regression or pivot operation just completed;

b) The degrees of freedom, Regression Sum of Squares (RSS) and Mean Sum of Squares (MSR);

c) The name of the standard and its relative concentration and standard error;

d) The partial F-value to remove and associated Q-value and x-value, and the new MSR for each standard.

e) If the next operation to be performed is a forward pivot, the name, partial F to add and associated Q- and x-values, the square root of the partial correlation coefficient, RHO, and the new MSR, are output for each standard being considered.

(9) FINAL RESULT

This section of output is broken down as follows:

a) "FIT"

The "FIT" is the reduced χ^2 statistic and gives an overall quantitative value for the fitting process. χ^2 is computed by the program as follows. The residual r_i in each channel (the difference between the sample spectrum and the least-squares spectrum) is computed: $\Sigma(y_i-f_i)$. The variance in channel i for the least-squares spectrum σ_i is then computed from the formula

$$\sigma_i = (f_i+0.1) + b_i t/t'(1+t/t'),$$

Then $$\chi^2 = \sum_i r_i/\sigma_i$$

and reduced χ^2 is $= (1/D_f)\sum_i r_i$

where D_f = number of (channels - standards).

b) "RESULTS + CONCENTRATIONS AND EST STANDARD ERRORS"

The nuclide name, nuclide concentration, and the estimated standard error is printed out for each nuclide, in the "printout" order. The concentration is in the same units as the standards in the nuclide library, and is the "actual" concentration determined by multiplying the "relative" concentration computed by the program multiplied by factors that take into account sample decay, volume reduction and result multiplication, the activity of the standard, and sample and standard counting times. The relative standard error is multiplied by the same factors. Also included in this section of output are the final partial F-values and the corresponding Q- and x-values for each standard in the final set.

c) "ANALYSIS OF VARIANCE"

This table lists the contributions of the regression and residuals to the total variance. In it are given the degrees of freedom and sums of squares for the regression, residuals and total; the mean sums of square due to the regression and residuals, and the overall F-value. The Mean Square Residual has an expected value for a good fit of $1.0 \pm (2/D_f)^{1/2}$.

(10) "LLNL R-STATISTIC"

This output is the result of the analysis of the residuals from the least-squares fit. It gives the auto-correlation statistic,[7] its expectation value, standard deviation and an approximate associated FAP. A low value of the auto-correlation coefficient means that the distribution of residuals is random. As this value approaches one (1) the residuals are becoming more correlated, indicating a poorer fit to the data.

(11) "FINAL STATISTIC FOR ELIMINATED STANDARDS"

This is a listing of the name and final partial F-value and associated Q- and X-value for each library standard eliminated from the set during the regression. It is useful to determine whether a standard was eliminated because it is truly uncorrelated with the sample or the threshold value for the F-test (variable Fø in UEDIT) was too high.

(12) "RATIO OF RESIDUALS OVER STD DEV PER CHANNEL"

The "residual" in each channel is the difference between the sample spectrum and the least-squares spectrum (y_i-f_i). The standard deviation for the least-squares spectrum is defined to be

$$\sigma_i = [(f_i + 0.1) + b_i t/t'(1+t/t')]^{1/2}$$

The program prints out one ratio for each channel in the least-squares spectrum; the printout should be read horizontally from left to right. These ratios, when squared, summed for all channels, and divided by the number of degrees of freedom D_f, will = "reduced" χ^2.

An examination of these ratios reveals where the least-squares spectrum f_i fits poorly the sample spectrum y_i, since these ratios will be "large." Ratios $\geq$ 10 or three consecutive ratios $\geq$ 2 are flagged as "suspicious channels" and printed out in the next part of the output.

(13) "SUSPICIOUS CHANNELS"

The location and contents of any suspicious channels are given. These are the channels where the ratio of the residual to the standard deviation, r_i/σ_i, is greater than 10 or the central channel of three consecutive channels that are all greater than +2. These are the same ratios that are printed out in the section "Residuals/Standard Deviation per Channel"; the "Suspicious Channels" indicates those channels where the least-squares fit is poor. The "suspicious channels" pointed out may or may not indicate the presence of an unexpected nuclide.

(14) "DISTRIBUTION OF RESIDUALS"

This is a histogram plot of the distribution of the residuals/standard deviation per channel. It is preceded by a moment analysis of the distribution of residuals giving the "MEAN," "VARIANCE," "SKEWNESS," and "EXCESS." The shape described by the moments as well as the visual identification of large residuals can be used in determining the "goodness of the fit" to the sample spectrum. In the present version of the subroutine RESID, residuals with $r_i/\sigma \geq 5$ are put in the first or last channel depending on the sign of the residual.

(15) "DISTRIBUTION OF RUNS"

This is a histogram plot of the number of positive and negative runs in the residual spectrum. A run is a series of consecutive channels in the residual spectrum. For a good fit, the expected value for the number of runs, for these spectra, is $D_f/2 \pm (D_f/4)^{1/2}$. The presence of large runs or a low number of runs can be an important aid in determining the randomness of the residuals and, hence, the goodness of the fit.

(16) "LOCATIONS AND SIZES OF LARGE RUNS"

The last item in the PREGA output is the location (or starting channel) of the large runs, and the length of the run. The sign of the length denotes whether this is a run of positive or negative residuals. In the present version of the subroutine RUNS only runs of absolute length five or greater are output.

VARIABLES (PREGA)

NAME (DIMENSIONS)	USAGE
A(10,10)	Square matrix to be inverted
AC(10)	Activity of standards from library
ARR(10,64)	Matrix containing 64 channels of (max.) 10 standard spectra; for residual calculation
ASC(60)	Disk locations of standard spectra
B(10)	$C_j = \sum_i w_i a_{ij} x_i$
BA(512)	Spectrum for either standard background or sample background, depending on location in program
BR	User option to do regression with trial set consisting of; = 0 all standards chosen = 1 background only NOTE: FOR BR=1; BACKGROUND MUST BE STANDARD No. 1 IN THE LIST
C	Bit pattern of standards currently in regression for present pivot
CC(10,10)	Correlation matrix for standards and sample
CH	x^2 value for least-square spectrum
CHDF	= CH/DN; "reduced" x^2 for least-squares spectrum

VARIABLES (PREGA) (Cont'd)

NAME (DIMENSIONS)	USAGE
D	Bit pattern of standards removed from original set
DAY	Decay time of sample (seconds)
DEF(10)	Hollerith constant DEF 8,
DF	Degrees of freedom; NCH-P
DG	Regression degrees of freedom; N
DN	Degrees of freedom; the number of channels used in the least-squares fit minus the number of standard isotopes
DR	Residual degrees of freedom; MF-NZ-P+1
DT	Total degrees of freedom; DG+DR
E	Bit pattern integer for standards currently in regression analysis
EF(10)	Logical variable to keep track of standard status in subroutine PIVOT
F	Bit pattern integer of original set of standard nuclei used
FØ	F-test rejection coefficient
FAT	t/t'; counting time of standard divided by counting time of standard background
FD	Correction for decay of sample and standard
FF	Final F-statistic = XG/XR
FILE	Logical unit number 8 (disk file:spectra)
FMAX	Maximum F-value (I) in forward pivot
FMIN	Minimum F-value (I) in backward pivot
FP(10)	F-value (I) after doing pivot
FPM(10)	F-value (I) for "F-to-remove" or "F-to-add"
FPS	Partial F-value sent to subroutine XQCALC
FS	t/t'; counting time of sample divided by counting time of sample background

VARIABLES (PREGA) (Cont'd)

NAME (DIMENSIONS)	USAGE
FX	t/t'(1 + t/t'); second order correction term for time difference
HA(10)	Half-life of standards from library: input library order
HAT(10)	Half-life of standards; printout order
HF(5)	Hollerith output variable for titles
HG	Hollerith output variable for titles
HH(6)	Hollerith output variable for titles
HR	Hollerith output variable for titles
HT	Hollerith output variable for titles
I	Index used for nuclide and channel number
Iø	Beginning of loop over standards value
I1	End of loop over standards value
IDAT	Associated variable for logical units 9 and 11
IFORKL(30)	Variable format statement number associated with ENCODE statement
IFORN(20)	Variable format statement number associated with ENCODE statement
II	Index used in library standards read/write operations
IN	Integer resulting from checking if standard I is in the current list before a pivot
INEG	Integer used to check sign of partial correlation coefficient (R2=RHO*RHO)
IPO	Integer used to direct program in next step after output of a pivot result
IR(512)	Suspicious channel number
IS(10)	Printout order of standards; OR in UEDIT can change as program is executed

VARIABLES (PREGA) (Cont'd)

NAME (DIMENSIONS)	USAGE
ISL(10)	Printout order of standards; OR in UEDIT never changes during execution
IT(10)	Serial order of standards to be used in least-squares fit
IVAR	Associated variable for logical units 8 and 10
J	Index used for nuclide and channel number
JJ	Index used with ASC for sample and background spectra
JJJ	Do loop counter
JDF	Standard set F minus standard set D, used at end of backward pivot
K	Index used for nuclide and channel number
KAC	Variable associated with ENCODE statement for format 916
KRO	Output residual (Yes = 1, No = 0)
LN	Number of standards used plus 1, N + 1
LOOP	Integer to read 64 channels from disk for residual calculation
LSK	Variable associated with ENCODE statement for format 916
LW	User option for weighting factor p_j; WF in UEDIT if based on actual counts = 1 for $p_j = 1$ = 0 for p_j = 1/(sample counts + background counts) = -1 for p_j = 1/(sample counts + 1.0)
M	Number of channels of standard and sample spectra (max = 512)
MF	Final channel number for computation
MI	Logical unit 9 (disk file: nuclide library)
MIU	Logical unit 11 (disk file: unknown sample)
MO	Logical unit 6 (line printer)

VARIABLES (PREGA) (Cont'd)

NAME (DIMENSIONS) USAGE

MP = 6 logical unit number of line printer

MY(10) Integer vector used to check bit pattern of current set of standards for a particular standard

N Number of standard nuclides to be fit (max = 9)

N2 Maximum channel number minus two; used for suspicious channels calculation

N5 Used to represent one (specified by program) value from IS or IT array

N6 Used to represent one (specified by program) value from IS or IT array

NBA User option for standard background subtraction; BS in UEDIT

= 1 for subtraction
= 0 for no subtraction

NBR User option for background; BG in UEDIT

= 1 for background supplied
= 0 for no background supplied

NBS User option for sample background subtraction
= 1 subtraction
= 0 no subtraction

NCH Number of channels used for calculation; MF-NZ+1

NDET = 0 dead time correction (ø = no, 1 = yes)

NEWST User option for application of rejection coefficient; RC in UEDIT
= 1 for application
= 0 for no application

NIT Carry over from old NAI program. To be eliminated when SEDIT and UEDIT are restructured

NNN = 0 counter index used in calling subroutine DECAY

NS Number of standard spectra in library (maximum = 40)

NW User option for weighting factor p_i; WD in UEDIT

VARIABLES (PREGA) (Cont'd)

NAME (DIMENSIONS)	USAGE
NZ	Initial channel used for calculation
P	Number of standards currently in regression
Q	Rejection coefficient; RC in SEDIT
QF	Q(X) associated with F-values
QH	User defined threshold channel number, TC in UEDIT
R(512)	1. Standard spectra (up to line 493) then 2. Ratio of residual divided by standard deviation
R2	Concentration of standard squared divided by standard error of standard divided by residual sum of squares; used to calculate partial F-values for PIVOT; partial correlation coefficient
RE	Residual count; RE = Y(J) - SV
RHO	Square root of partial correlation coefficient, R2
RMSR	Residual sum of squares divided by DF
RSS	Residual sum of squares
RT	x^2 sum for least-squares spectrum
S(512)	Standard spectra counts; a_{ij}
S1	$= \sum_i x_i$; sum of counts for the sample spectrum
S2	S1 + SB * FX; sum of sample plus corrected background counts
SA	b_{jk}; least square sum coefficients for unknown concentrations Z(I) and sample Y(I); store in A(L,K); total sum of squares
SB	Sum of counts for the sample background spectrum from the initial to the final channel
SG	Regression sum of squares
SIR(50)	Contents of suspicious channels
SNAM(2)	Sample file name used in output for identification

VARIABLES (PREGA) (Cont'd)

NAME (DIMENSIONS)	USAGE
SQUC	Used to check sign of number before taking square root
SR	Residual sum of squares
SRD1	σ_{II} for correlation coefficient CC(I,J)
SRD2	σ_{JJ} for correlation coefficient CC(I,J)
SS(10)	$\sum_i a_{ij}t_j$; sum of counts for standard isotopes from the initial to final channel
ST	Residual sum of squares
STD(10)	1. Variance of the nuclide concentration; input library order 2. Corrected standard error of the nuclide concentration; input library order
SUM	Relative nuclide concentration, m_j
SUM W	Sum of weighting factors, ρ_i, from initial channel to final channel
SV	Least squares spectrum sum, $\sum_i f_i$
SX	c_j, least squares sums for constant terms
T	1. Y(I)*BA(I)*FX defined at program line 2. Reciprocal of p_i defined at program line 3. r^2; residual squared defined at program line 480
TB	Counting time for background
TE	1. Y(J)*BA(J)*FX used in calculation of CHDFS
TIS1	Hollerith constant to store "SAMP"
TIS2	Hollerith constant to store "LE "
TISO(20)	Names of stored nuclides used in least squares fit; in printout order
TISOT(40)	standard nuclide names; input library order
TMO	= (0.1 + SV); reciprocal of p_i based upon least-squares spectrum

VARIABLES (PREGA) (Cont'd)

NAME (DIMENSIONS)	USAGE
TMP	= TMO + BA(J)*FX; reciprocal of p_i based upon least-squares spectrum + corrected background; defined at program line 484. Redefined at program line 493 as the square root of TMP.
TNAME(36)	Original location of sample spectrum
TSA	Counting time for sample in seconds
TST(10)	Counting time of standard in seconds; from PREGA library
VM	User defined result multiplication factor; MF in UEDIT
VRED	User defined volume reduction factor; VR in UEDIT
VU	Sum of residuals squared
VVV	Sum of TMP
VY	Residual squared times p_i
W(512)	Weighting factors p_i
XF	X-value returned from subroutine XQCALC and checked against FØ
XG	Final regression sum of squares divided by degrees of freedom
XR	Final residual sum of squares divided by degrees of freedom
Y(513)	Sample spectrum X_i
Z(10)	Library nuclide concentration
ZS	Nuclide concentration sent to subroutine XQCALC

Program Calculations

PREGA is a FORTRAN language program consisting of approximately 638 lines and requiring ~ 52K words of memory for execution. The main calculation steps used by the program to calculate the best subset of standards using the pivotal regression technique, the nuclide concentration, residuals and tne statistical analysis of accidental errors are listed on the following pages. This section is included to provide further understanding of the calculation and as a guide for future modifications that may become necessary or desirable.

PREGA Program Calculations

Program Line Number	FORTRAN Expression	Description	Equation Number
70	FAT = TST(I)/TB	Counting time of standard i counting time of background	1
72	S(J) = S(J)-BA(J)*FAT	[Standard counts - corrected background counts] in each channel summed over all channels	2
103	FS = TSA/TB	Counting time of sample counting time of background	4
104	FX = FS + FS**2	Second order correction factor used in line 484	5
121	Y(I) = Y(I)-BA(I)*FS	[Sample counts - corrected *background* counts] in each channel i	6
124	SB = SB + BA(I)	Total background counts; this is the "BACKGD SUM" that is printed out	7
129	S1 = S1 + Y(I)	Total sample counts (corrected background subtracted); this is the "SAMPLE SUM" that is printed out	8
211	SS(L) = SS(L)+R(J)/W(J)	Weighted sum of standard counts in all channels i for nuclide j; this is the "WEIGHTED SUM OF STANDARDS AND UNKNOWN" that is printed out	3

SET UP ARRAY A(N+1, N+1) AND CORRELATION MATRIX

PREGA Program Calculations (Cont'd)

Program Line Number	FORTRAN Expression	Description	Equation Number
216	S(J) redefined as original standard spectra	FORTRAN S(J) here represents the standard spectra	
219	SA=SA+S(I)*R(I)*W(I)	Standards cross product terms times weighting factor stored in A(L,K)	9
225	SA=SA+R(I)*Y(I)*W(I)	Cross product of standards and sample terms stored in A(N+1,L) and B(L)	10
233	SA=SA+Y(I)*Y(I)*W(I)	Stored in A(N+1, N+1); this location is the residual sum of squares during the pivot operations. Also stored in ST	11
244	SRD1=A(K,K)-SS(N6)**2/SUMW	Variance of standard K (or sample)	12
245	SRD2=A(L,L)-SS(N5)**2/SUMW	Variance of standard L (or sample)	13
246	CC(K,L)=(A(K,L)-SS(N6)*SS(N5)/SUMW) /SQRT(SRD1*SRD2)	Correlation coefficient: covariance of standards K and L divided by the square root of the product of their variances	14
START PIVOTAL REGRESSION ANALYSIS			
329	FPM(I)=(NCH-P)*((A(I,LN)**2)/ (-A(I,I)*A(LN,LN)))	Calculate partial F to remove; NCH-P = degrees of freedom	15

PREGA Program Calculations (Cont'd)

Program Line Number	FORTRAN Expression	Description	Equation Number

381	FPM(I)=((NCH-P-1)*R2)/(1.0-R2)	Calculate pal F to add;
		NCH-P-1 =degrees of freedom
			16

409	RSS = A(LN,LN)	Residual sum of squares	
410	RMSR = RSS/DF	Residual sum of squares per degree of freedom	
418	Z(J) = -A(I,LN)*TST(N5)/TSA	Relative concentration of standard J	17
422	STD(J) = SQRT(SQUC)*RMSR) *TST(N5)/TSA	Relative standard error of standard J determined from the variance	18
448	SG=SG-A(I,LN)*B(I)	Regression sum of squares	
475	SV = SV+ARR(JJ,JJJ)*(-A(I,LN))	Calculates the least-squares spectrum counts in each channel (because of memory limitations, this is done 64 channels at a time)	19
479	RE = Y(J)-SV	Residual = difference between sample and least-squares spectrum	20
480	T = RE**2	Squares the residual	21
481	VY = VY+W(J)*T	Residual sum of squares; stored in SR at line 472	22
482	VU = VU + T	Sum of residuals squared over all channels	23
483	TMO = ABS(0.1 + SV)	f_i must $\neq 0$; zero divide prevention	24

PREGA Program Calculations (Cont'd)

Program Line Number	FORTRAN Expression	Description	Equation Number
484	TMP = TMO + BA(J)*FX		25
485	VVV = VVV + TMP	Accumulates TMP over all channels	26
491	RT = T/TMP	Residual squared TMP	
492	CH = CH + RT	Accumulates RT over all channels; this is chi-squared for least-squares spectrum	27
493	TMP = SQRT(TMP)	Redefine TMP as square root of TMP	
494	R(J) = RE/TMP	Calculates residuals over standard deviations; printed out as "RATIO OF RESIDUALS OVER STD DEVIATION PER CHANNEL"	
498	DN = MF-P-NZ+1	Calculates the number of degrees of freedom for χ^2, MF = end channel, NZ = start channel, P = number of standards in fit	
501	DT = DR+DG	Total number degrees of freedom	
502	CHDF = CH/DN	"Reduced" χ^2 for the least-squares spectrum; this is the "FIT" that is printed out	
513	FD = EXP(0.693*DAY/HA(N5))	Correction for decay of sample DAY = $t_{1/2}$=half-life of sample and HA = $T_{1/2}$=half-life of standard	28
514	STD(J)=SQRT(-A(I,I)*A(LN,LN)/DN) *TST(N5)/TSA	Relative standard eror determined from variance of standard 1	

PREGA Program Calculations (Cont'd)

Program Line Number	FORTRAN Expression	Description	Equation Number
515	STU(J)=AC(N5)*FD*STD(J)*VM/VRED	Corrected standard error; AC is the activity of the sample; VM is result multiplication factor; VRED is volume reduction factor	
516	Z(J)=(-A(I,LN)*(TST(N5)/TSA) *AC(N5)*FD*VM/VRED	Relative concentration m_j is converted to actual concentration level	
526	XG = SG/DG	Reduced regression sum of squares	
527	XR = SR/DR.	Reduced residual sum of squares	
528	FF = XG/XR	Final F-statistic	

Equations Used in PREGA (FORTRAN variables are underlined)

1. $\underline{FAT} = t_j/t'$

2. $\underline{S(J)} = \sum a_{ij} = \sum S_{ij} - b_i(t_j/t')$

3. $\underline{SS(J)} = \sum a_{ij}$

4. $\underline{FS} = t_s/t'$

5. $\underline{FX} = (t_s/t')(1 + t_s/t')$

6. $\underline{Y(I)} = x_i = z_i - b_i(t_s/t')$

7. $\underline{SB} = \sum b_i$

8. $\underline{S1} = \sum x_i$

9. $\underline{A(L,K)} = \sum a_{ik} a_{il} w_i$

10. $\underline{SA} = \sum a_{il} x_i w_i$

11. $\underline{SA} = \sum x_i x_i w_i$

12. $\underline{SRD1} = \sigma_{kk} = A_{kk} - (\sum a_{ik} w_i)^2 / \sum w_i$

13. $\underline{SRD2} = \sigma_{ll} = A_{ll} - (\sum a_{il} w_i)^2 / \sum w_i$

14. $$\rho_{kl} = \frac{A_{kl} - \left[\sum_i a_{ik} w_i \sum_i a_{il} w_i / \sum_i w_i \right]}{(\sigma_{kk}\ \sigma_{ll})^{1/2}}$$

15. $F_p^- = DF*R^2 = DF*A_{I,N+1}^2 / (-A_{II} A_{N+1,N+1})$

Equations Used in PREGA (FORTRAN variables are underlined) (Cont'd)

16. $F_p^+ = DF*R^2/(1-R)^2$

17. $\underline{Z(J)} = \sum_i a_{ij}*t_j/t' = m_j$

18. $\underline{STD(J)}$ $(A-(I,I)*A(N+1,N+1)/DF)t_j/t'$

19. $\underline{SV} = f_i = \sum_j a_{ij}m_j$

20. $r_i = (x_i - f_i)$

21. $r_i^2 = (x_i - f_i)^2$

22. $\underline{VY} = \sum_i r_i^2 w_i$

23. $\underline{VU} = \sum_i r_i^2$

24. $\underline{TMO} = 0.1 + f_i$

25. $\underline{TMP} = (0.1 + f_i) + b_i \underline{FX}$

26. $\underline{VVV} = \sum (0.1 + f_i) + b_i \underline{FX}$

27. $\underline{CH} = \sum r_i^2 / (0.1 + f_i) + b_i \underline{FX}$

28. $\underline{FD} = \exp(0.693 t_{1/2}/t^s_{1/2})$

Software Organization

(1) Logical Unit and Disk File Structure

The program PREGA and the two editor programs SEDIT and UEDIT have access to data stored in disk files via specific, user-defined logical units. The organization of these programs, disk files, and logical units is shown in figure 2.

The program PREGA uses logical units 6, 8, 9, 10, and 11. It shares logical unit 9 with the SEDIT editor program and shares logical unit 11 with the UEDIT editor program.

Logical unit 6 is used to print out the results of the program; it is usually a line printer.

Logical unit 8 is a disk file that stores spectral data that have been accumulated by the user in the laboratory. The program PREGA reads this data to perform its calculations. Each spectrum is stored in one disk file element. The data will include the sample spectrum, the background spectrum for the sample, the standard spectra (up to 9 standards are permitted at present) and the background spectrum for the standard. The background spectrum must be standard No. 1 in the library.

Logical unit 9 is the standard nuclide library file. The file contains information common to all nuclides in the library (the file "header") as well as specific information about each nuclide (the "list of standards"). For each standard spectrum in logical unit 8 there will be a corresponding entry in the nuclide library file. The editor program SEDIT is used to add, delete, or make changes in this file.

Logical unit 10 is used by the program PREGA for temporary storage of large data arrays. This is a "scratch" file.

Logical unit 11 is a disk file that contains information about the unknown spectrum the user has acquired. The UEDIT editor program is used to add, delete or change the information in this file.

The file structures for logical units 8, 9, 10, and 11 are shown on the following pages. The file structures for logical units 9 and 11 show two columns of program parameter names. The editor programs use the names in the first column and the program PREGA uses the names in the second column.

The following relation exists among sectors, records, words, bytes, and bits for all files defined as logical units 8, 9, 10, and 11 in the program PREGA:

1 sector = 64 records = 128 words = 256 bytes
1 record = 2 words = 4 bytes = 32 bits
1 word = 2 bytes
1 byte = 8 bits

(1a) Logical Unit 8

The program PREGA uses the GET subroutine to read each spectrum through logical unit 8. Each file element contains one spectrum. The spectral data begins with record number 129 (decimal byte offset 512); the first 128 records (511 bytes) in each file element are not used by PREGA. The first 128 records comprise the Header section of Nuclear Data spectral files and contain information used by other programs (e.g. GSHIFT, GET, PUT).

The numbering system is illustrated below for the first 2 records.

Word 2	Word 1	Record 129
byte 515 byte 514	byte 513 byte 512	
Word 2	Word 1	Record 130
byte 519 byte 518	byte 517 byte 516	

Data type coding is as follows:

Sptr (Spectral data) The spectral data from each channel is stored in 4 bytes. Only 30 bits are used. The values are stored as double integers. Word 1 is the least significant word.

Regardless of the number of bits/channel assigned to an ADC, each channel is always written to disk as a 32-bit word.

If M = the number of channels of spectral data, the decimal byte length of the file element is 512 + 4*M. The user can use this to determine how much disk space must be reserved for the spectral data. For example, the maximum number of channels a user can have in one spectrum is 512. This would require 2560 bytes or 10 sectors to be reserved for that file element. The maximum number of spectra accessible by SEDIT at any given time is currently 9. The number of spectra which can be stored on the disk in file DATA is determined by the size of file DATA. Because SEDIT is limited to 9 standards, the maximum number of spectra, or file elements a user can have on logical unit 8 is 12 (see below) so the upper limit of the size of this disk file is 120 sectors.

The total number of file elements will be the sum of:

1 to 9 file elements for the standard spectra
1 file element for the standard background spectrum
1 file element for the unknown sample spectrum
1 file element for the unknown sample background spectrum

NOTE - The program SEDIT can actually handle up to 40 standard spectra. The limitation of only 9 standard spectra is caused by PREGA. As stated earlier, PREGA was built on the framework of the Nuclear Data, Inc. NAI analysis program, which has some built-in inefficiency. The method for modifying PREGA and, hence the whole package is given in more detail below. At present, it was felt that 9 standards would be sufficient for the present analysis without making PREGA unnecessarily larger.

Structure of PREGA File 8

Record Number	Decimal Byte Offset	Decimal Byte Length	Parameter Description	Data Type
1	0	4	Not used	...
2	4	4	Not used	...
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
128	508	4	Not used	...
129	512	4	Number of counts in channel 1	Sptr
130	516	4	Number of counts in channel 2	Sptr
131	520	4	Number of counts in channel 3	Sptr
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
641	2560	4	Number of counts in channel 512	Sptr

(1b) Logical Unit 9

SEDIT writes the file to logical unit 9 and the program PREGA reads from logical unit 9.

The numbering system is illustrated below for the first 2 records.

Word 2	Word 1	Record 1
byte 3 byte 2	byte 1 byte 0	
Word 2	Word 1	Record 2
byte 7 byte 6	byte 5 byte 4	

Data type coding is as follows:

A	ASCII characters
F	REAL*4 single precision floating point with 4 bytes
I*2	INTEGER*2 with 2 bytes
L*1	LOGICAL*1 with 1 byte

The number in the "offset" column is the address.

This file requires 256 bytes for the "header" section and 64 bytes for each nuclide in the library. If N represents the number of nuclides in the library, then a total of 256 + 64N bytes of storage are required; one sector = 256 bytes. For example, the maximum number of nuclides a user can have in the library is 9. This would require 256 + 576 = 832 bytes. An example printout of this file is given in the section about SEDIT. The example consists of two parts: the first part is the "PREGA LIBRARY HEADER" and the second part is the "PREGA LIBRARY STANDARDS".

Structure of PREGA and SEDIT File 9

Record Number Offset	Decimal Byte Offset	Decimal Byte Length	SEDIT Name	PREGA Name	Parameter Description	DATA Type
1	0	2	M	NS	number of isotopes in standard library (max = 9)	I*2
2	4	2	NC	M	number of channels per standard (max = 512)	I*2
3	8	2	IT	NIT	number of iterations for threshold and gain calculation (not used)	I*2
4	12	2	BS	NBA	background subtract 1 = yes, ø = no	I*2
5	16	2	IS	NZ	initial channel for computation	I*2
6	20	2	IE	MF	final channel for computation	I*2
7	24	132			space reserved for file expansion	
40	156	4	BT	TB	counting time for standard background	F
41	160	4	RC	Q	rejection coefficient	F
42	164	30	BA	ASC	standard background location	L*1
49	194	2			word 2 of record 49 not used	
50	196	60			space reserved for file expansion	
65	256	8	A	TISOT	first standard isotope name	A
67	264	30	A	ASC	first isotope location	L*1
74	294	2			word 2 of record 74 not used	
75	296	4	HL	HA	half-life of first isotope	F
76	300	4	SEC	TST	counting time of first isotope	F
77	304	4	ACT	AC	activity of first isotope	F
78	308	12			not used	
81	320	52			data for standard isotope two	
94	372	12			not used	
97	384	52			data for standard isotope three	
110	436	12			not used	

.
.
.

through 65+ (NS*16) -3
where NS = number of isotopes in
standard library

The second word of records 1,2,3,4,5 and 6 is not used.

(1c)   Logical Unit 10

PREGA reads and writes to the file of the logical unit 10. This file, a "scratch" file, is used for temporary storage of the large data arrays manipulated by the NaI program.

The numbering system is illustrated below for the first 2 records.

Word 2	Word 1	Record 1
byte 3 byte 2	byte 1 byte 0	
Word 2	Word 1	Record 2
byte 7 byte 6	byte 5 byte 4	

Data type coding is as follows:

Sptr (Spectral Data) The spectral data from each channel is stored in 4 bytes. Only 30 bits are used. The values are stored as double integers. Bits 14 and 15 have been set for ϕ for word 2, which is the most significant word.

Structure of PREGA File 10 ("Scratch" file)

Record Number Offset	Decimal Byte Offset	Decimal Byte Length	PREGA Name	Parameter Description	DATA Type
1	0	4*M	S	storage for all channels of first standard spectrum	Sptr
M+1	4*M	4*M	S	storage for all channels of second standard spectrum	Sptr
2*M+1	2*4*M	4*M	S	storage for all channels of third standard spectrum	Sptr
3*M+1	3*4*M	4*M	S	storage for all channels of fourth standard spectrum	Sptr
.	.	.	.		
.	.	.	.	.	
.	.	.	.	.	
NS*M+1	NS*4*M	4*M	S	storage for all channels of sample spectrum	Sptr

where M = number of channels per standard

NS = number of isotopes in standard library

(1d) Logical Unit 11

UEDIT writes the file to logical unit 11 and PREGA reads from logical unit 11.

The numbering system is illustrated for the first 2 records.

Word 2	Word 1	Record 1
byte 3 byte 2	byte 1 byte 0	
Word 2	Word 1	Record 2
byte 7 byte 6	byte 5 byte 4	

Data type coding is as follows:
F REAL*4 single precision floating point with 4 bytes
I*2 INTEGER*2 with 2 bytes
L*1 LOGICAL*1 with 1 byte

The number in the "offset" column is the address.

An example of the printout of this file is given in the section on UEDIT with the heading "PREGA UNKNOWN SAMPLE PARAMETERS".

Structure of PREGA and UEDIT File 11

Record	Decimal	Decimal				
Number	Byte	Byte	UEDIT	PREGA		DATA
Offset	Offset	Length	Name	Name	Parameter Description	Type
1	0	4			dummy	
2-9	1	30	UN	ASC	location of sample spectrum	L*1
9	34	2			word 2 of record 9 not used	
10	36	4	GN	FIT	gain shift ratio (not used)	F
11	40	4	TH	SHCT	threshold shift (not used)	F
12	44	2	BG	NBR	background supplied yes = 1 no = ϕ	I*2
13	48	2	BS	NBS	background subtract yes = 1, no = ϕ	I*2
14	52	4	BT	TB	counting time for sample background (sec)	F
15	56	4	CT	TSA	counting time for sample (sec)	F
16	60	4	VR	VRED	sample concentration factor	F
17	64	4	DT	DAY	decay time (sec)	F
18	68	4	MF	VM	sample dilution factor	F
19	72	30	BA	ASC	location of sample background	L*1
27	104	2			word 2 of record 27 not used	
28	108	2	WD	NW	weighting factor determination: = ϕ if based on actual counts/channel = 1 if based on calculated counts/channel	I*2
29	112	2	NS	N	number of library spectra to be fitted to the sample	I*2
30	116	2	Fϕ	Fϕ	F-test rejection coefficient	R*4
31	120	2	WF	LW	weighting factor = -1 for (sample counts +1) **-1 = ϕ for (sample counts + background counts)**-1 = 1 for 1	I*2
32	124	2	RC	NEWST	apply rejection coefficient	I*2
33	128	2	BR	BR	start regression with background only; ϕ = no, 1 = yes	I*2
34	132	2	BP	BP	allow pivot on background ϕ = no, 1 = yes	I*2
35	136	2	RO	KRO	output residuals ϕ = no, 1 = yes	I*2
36	140	32	RS	ASC	ID of output file for residuals	L*1
43	176	4*N	OR	IS	library standard number listed in the order of desired printout	I*2

N is the number of library nuclides to be fit to the sample.

The second word of records 12,13,27,28,29,30,32,33,34, and 35 through 43+N is not used.

Subroutines Called by the Program PREGA

The following eleven subroutines are used during the execution of PREGA. Figure 3 shows the hierarchy of subroutine calls. Brief descriptions of each subroutine are given below. Listings of the subroutines are given in Appendix B.

(1) GET+(ASC, ARRAY, IVAR, FILE, M)

The subroutine GET is used by the NAI program to obtain spectral data from the various disk file/elements defined as logical unit 8. The program PREGA first obtains the location of the spectrum and sends this location, as an ASCII character string, to the GET subroutine. GET in turn calls CONCAT which concatenates the character string and LUNDEF subsequently sends the concatenated string to the operating system, MIDAS, which interprets the string and performs the command operation. GET is a FORTRAN language subroutine, while the other two, CONCAT and LUNDEF, are written in assembler language.

Listings or descriptions of the system routines CONCAT and LUNDEF are given in Appendix C.

VARIABLES (GET)

NAME (DIMENSIONS)	USAGE
ARRAY(256)	Channel Contents of Spectrum Being Retrieved from Disk
ASC(60)	Location of spectrum being retrieved (FILE. ELEMENT (, DEVICE))
DEF(10)	Hollerith constant DEF8, used to create a command in CONCAT and LUNDEF
FILE	Logical unit number 8 (disk file:spectra)
IVAR	Associated variable for logical unit 8
K	Loop index over number of channels in spectrum
LSB	Least significant byte of data word
M	Number of channels in spectrum
MSB	Most significant byte of data word
NAM(60)	Concatenated character string ASC sent to LUNDEF
RE1	Most significant byte of data word ARRAY(K) converted to R*4
RE2	Least significant byte of data word ARRAY(K) converted to R*4

(2) HISTO (LABEL, IH, XL, XD, N)

The FORTRAN subroutine HISTO is called by the Subroutines RESID and RUNS. From the data received from these two subroutines it generates a labeled, scaled histogram which is output to logical unit 6. The two histograms are "DISTRIBUTION OF RESIDUALS" and "DISTRIBUTION OF RUNS". This routine was written by G. Phillips at NRL in June of 1981.

VARIABLES (HISTO)

NAME (DIMENSIONS)	USAGE
A(50)	Data point character '*'
ASTAR	Data point character '*'
ACOLON	Data point character ':'
I	Loop index to skip 5 lines after histogram is output
IDIV	Scale factor for output histogram
IH(21)	Number of steps on x-axis (21 MAX)
IOUT	Logical unit = 6 for output
IT	$\sum x_i$ for output histogram
J	Y-value for each channel x
K	Loop over number of x-values in histogram
LABEL (4)	Title label for output histogram
MAX	Maximum Y-value allowed; used to determine scale factor (MAX/IDIV $\le$ 50)
N	Number of x-values in histogram
X	First (or lowest) x-value
XD	Step size of x-values
XL	Lowest x-value from calling program

(3) PIVOT (A,P,E,K)

PIVOT is a FORTRAN subroutine called by PREGA. Its purpose is to produce the negative of the upper triangle of the inverse of a square symmetric matrix by operating only on the upper triangle of the matrix. The procedure by which this is accomplished is an adaptation of the Gauss-Jordon Method. On each entry to PIVOT a specified diagonal element is used for one pivot operation. To invert a k^{th} order matrix the procedure must be called k times. The order of which diagonal element is pivoted is unimportant. This routine checks for a near-zero pivot. All pivots are done with single-precision arithmetic with no significant loss of accuracy. The routine was written in FORTRAN by G. Phillips and B.G. Glagola, NRL in July 1982 from Algorithm AS 37 by M.J. Garside.[3]

VARIABLES (PIVOT)

NAME (DIMENSIONS)	USAGE
A(10,10)	Array of standards and sample to be inverted
AA	Reciprocal of the element about which the pivot occurs (A(P,P))
AIP	A(I,P)*AA or A(P,I)*AA
E(10)	Variable to determine which standards are in the current subset
I	Loop index over standards with array index less than that of the pivoting standard (max P1 = P-1)
J	Loop index over standards and sample with array index greater than that of the pivoting standard (min P2 = P+1)
K	Number of standards in original set plus one (in PREGA K = LN = N+1)
P	Index of the pivoting standard
P1	P1 = P-1; loop maximum
P2	P2 = P+1; loop minimum

(4) PUT (ASC, ARRAY, IVAR, FILE, M)

The FORTRAN subroutine PUT was written by G. Phillips, NRL in June 1981. PREGA uses PUT to write the residual spectrum on disk using logical unit 8. This routine also uses CONCAT and LUNDEF to send a character string to the MIDAS operating system and perform the command.

VARIABLES (PUT)

NAME (DIMENSIONS)	USAGE
ARRAY (256)	Spectrum to be output to disk
ASC (60)	Disk location of spectrum (File. Name (, Device))
DEF (10)	Hollerith constant DEF 8, used to create a command in CONCAT and LUNDEF
FILE	Logical unit number 8
IVAR	Associated variable for logical unit 8
K	Loop index over number of channels in output spectrum
LSB	Least significant byte of data word ARRAY(K)
M	Number of channels in output spectrum
MSB	Most significant byte of data word ARRAY(K)
NAM	Concatenated character string ASC sent to LUNDEF
V	Data word ARRAY(K)

(5) RESID (R, NZ, MF)

The FORTRAN subroutine RESID was written by G. Phillips, NRL in June 1981. It is used by PREGA to generate a distribution of residuals from the residual/standard deviation spectrum and perform a moment analysis of this distribution. The output from RESID is the "MEAN RESIDUAL", "VARIANCE", "SKEWNESS", and "EXCESS" of the distribution. RESID also calls HISTO to generate and output the histogram of the distribution of residuals.

VARIABLES (RESID)

NAME (DIMENSIONS)	USAGE
A(10)	Moments of the residual distribution
C(10)	Constants for the moment calculations
EX	Excess of the residual distribution
I	Index for loop over the number of channels in the residual spectrum
IH(21)	Number of steps on the x-axis for SUBROUTINE HISTO
IOUT	Logical unit for output = 6
J	Index for the loop to calculate C_j
K	Index for yield at each step on x-axis for HISTO
LABEL (4)	Title for histogram sent to HISTO
M	Number of channels used in calculation plus 1
MF	Initial channel used in calculations
N	Number of data points sent to HISTO
NZ	Final channel used in calculations
R(256)	Residual/standard error spectrum
SD	Second moment of residual distribution
SQ	Third moment of residual distribution, skewness
U	Mean residual
U2	Mean squared residual
U3	Mean cubed residual
U4	Mean fourth power residual
V	Variance of residual distribution
XD	Step size for HISTO
XL	Lowest x-value for HISTO

(7) RSTAT (NCH, P, R, NZ, MF)

The subroutine RSTAT analyzes the residual/standard deviation spectrum from PREGA to compute an auto-correlation coefficient, its expectation value, standard deviation and an approximate associated FAP. RSTAT uses an entry to the subroutine XQCALC to calculate the associated FAP. This version of RSTAT is an adaptation to PREGA by B.G. Glagola, NRL, 20 July 1982 of the original routine written by T.B. Gosnell, LLNL, 22 June 1982.

VARIABLES (RSTAT)

NAME (DIMENSIONS)	USAGE
CH	Number of channels in residual spectrum
DF	Number of degrees of freedom
EP	Number of standards in final set
EXRES	Expectation value of residual auto correlation coefficient
I	Index for loop over number of channels minus one
MF	Final channel in spectrum
MF1	MF-1
NCH	Number of channels in spectrum from PREGA
NRMR	Equivalent normal standard deviations
NZ	Initial channel in spectrum
P	Number of standards in final set from PREGA
Q	Residual statistic "False Alarm Probability" returned from XQCALC
R(512)	Residual/standard error spectrum from PREGA
RSTT	Auto-correlation coefficient
SIGRES	Standard deviation of R-statistic

(8) RUNS (R, NZ, MF)

The FORTRAN subroutine RUNS analyzes the residual/standard deviation spectrum from PREGA for runs of consecutive channels with the same sign. It outputs the location (starting channel of a run) and the size of the run. The sign of the size indicates a run of positive or negative residuals. RUNS also calls HISTO to generate and output a histogram of the distribution of runs. This routine was written by G. Phillips, NRL, June, 1981.

VARIABLES (RUNS)

NAME (DIMENSIONS)	USAGE
I	Index used for channel counting
ICH	Beginning channel of a particular run
IH(21)	Counter for the number of times a particular run length occurs
J	Output index over location and number of large runs
K	Index for x-axis of number of runs histogram
KSI(12)	Storage variable for length of long runs
LABEL(4)	Identification label for histogram, sent to HISTO
MF	Last channel of spectrum used for calculations
N	Number of data points sent to HISTO
NCH(50)	Initial channel of long runs for output
NRS(50)	Length of long runs for output
NZ	Initial channel of spectrum used for calculation
R(256)	Residual/standard error spectrum from PREGA
XD	Step-size of x-axis of output histogram
XL	Lowest value on x-axis of output histogram

Future Modifications of PREGA

The program SEDIT, as received originally from Nuclear Data, Inc., is capable of handling up to 40 standard nuclides in the library. At the present PREGA limits the number of standards that are allowed to be in the library. This is because PREGA was built on the framework of the Nuclear Data, Inc. NAI Analysis Package. In this package, the analysis program (called PREGA now) reads control parameters first from SEDIT and then from UEDIT. Hence, PREGA reads all of the standards listed by SEDIT from disk, and computes their sums and does background corrections before it reads from UEDIT which standards and in which order they are to be used in the regression analysis. To keep PREGA from becoming unnecessarily large, it was decided that nine standards would be sufficient to fit a sample spectrum. To increase the maximum number of standards allowed several variable dimensions in PREGA and subroutine PIVOT have to be modified. Currently these have the dimension (10) or (10,10) for arrays. These variables are listed below along with their identification, with N = maximum number of standards in the library.

PREGA

A(N+1, N+1)	The matrix that the pivot operations are carried out on
AC(N)	Activity of standards from library in SEDIT
ARR(N,64)	Matrix containing 64 channels of N standard spectra for residual calculation
B(N)	$C_j = \sum_i w_i a_{ij} x_i$
CC(N+1, N+1)	Correlation matrix for standards and sample
EF(N)	Logical variable to keep track of standard status in subroutine PIVOT
FP(N)	Partial F-value for output after pivot
FPM(N)	Partial F-to-add or F-to-remove
HA(N)	Half-life of standards from library in SEDIT
HAT(N)	Half-life of standard; printout order
IS(N)	Printout order of standards; OR in UEDIT can change during execution
ISL(N)	Printout order of standard; never changes during execution
IT(N)	Serial order of standards to be used in least-squares fit

PREGA (Cont'd)

MY(N)	Integer vector used to check bit pattern of current set of standards for a particular standard; NOTE - values must also be added to DATA statement for MY
SS(N)	Sum of counts over channels of standard spectra
STD(N)	Standard error of nuclide concentration
TISO(2*N),TISOT(2*N)	Names of standard nuclides
TST(N)	Counting time of standard in seconds, from SEDIT
Z(N)	Standard nuclide concentration

NOTE: For neatness of output, if the number of standards is increased the I/O statements for the correlation matrix at PREGA lines 204-208 and 248-252 must also be modified.

PIVOT

A(N+1, N+1)	Matrix to be pivoted
E(N)	Logical variable to determine which standards are in the current subset

8. PROGRAM SEDIT

Operation

The standards editor program, SEDIT, is used to prepare a file that contains control parameters for the standards and the standard library for use by PREGA. The library contains information about the name, disk location, half-life, etc., about each standard spectrum. SEDIT writes to a file assigned to logical unit 9. SEDIT allows the user to build, change or list the standard control parameters, and to also create a new library, insert or delete a standard, append new standards or list the present library. The structure of the output file on logical unit 9 is given in the section under PREGA software organization. A listing of SEDIT is given in Appendix B.

LANGUAGE

The program is written in DEC RT-11 FORTRAN.

INPUTS

Keyboard
Logical unit 5, all operator interaction and input is from keyboard

Logical Unit 11
Output file of UEDIT used for reading number of standards wanted for fit, for one standard library listing option

Logical Unit 9
For modifying an existing SEDIT output file

OUTPUT

Logical Unit 6
Record of operator interaction with SEDIT. This is usually defined as the keyboard unit the operator is using for input

File.Element
Logical unit 9 output file for SEDIT saved on disk for use with PREGA

A sample copy of the output from SEDIT is given in Appendix A.

VARIABLES (SEDIT)

NAME (DIMENSIONS)	USAGE
A(32)	Isotope name or file.element disk location
ACT	Activity of standard
ANS	Alphanumeric answer to SEDIT prompt (e.g. Y, N)
BA	Background spectrum location FILE.ELEMENT (,DEVICE)
BB	= B, for build new header containing standard control parameters
BS	Background subtract option
BT	Counting time for background spectrum in seconds
CC	= C, to change one of the header parameters
D	Days, half-life unit
EDT	Program pointer
FNUM	REAL*4 I/O variable
H	Hours, half-life unit
HL	Standard half-life
I	Index for program control
IE	Final channel for computation
INT(26)	Used to rewrite library when inserting or deleting a standard
IS	Initial channel for computation
IT	Number of iterations NOTE - no longer used, set = 1
IVAR	Record number for direct-access I/O
K	Check time units of half-life and program pointer
L	Defines logical unit for output
LA	Switch program to append library mode
LD	Switch program to delete standard mode

VARIABLES (SEDIT), (cont'd)

NAME (DIMENSIONS)	USAGE
LENGTH	Check for answer being carriage return only (to switch from header to library section or exit program)
LI	Switch program to insert standard mode
LN	Switch program to create a new library mode
LP	Switch program to list mode
LUI	Logical unit to read UEDIT variables NR and OR
LU	Logical unit for output file of SEDIT
M	Total number of standards in library
MLA	NS from UEDIT for list standard option
MLI	OR from UEDIT for list standard option
MM	Months, half-life unit
N	Counter for number of standards in library
NC	Number of channels per standard
NUM	I*2 I/O variable
RC	Rejection coefficient no longer used, set = 1
SEC	Counting time of standard in seconds
Y	YES, I/O answer to SEDIT prompt

9. PROGRAM UEDIT

Operation

The unknown sample editor program, UEDIT, is used to prepare a file that contains control parameters and information about the sample for use by PREGA. UEDIT writes to a file assigned to logical unit 11. The program SEDIT also reads the UEDIT output file to control an operator option for listing the standard library. UEDIT allows the user to Build, Change, or List the output file with prompting questions. The structure of the output file on logical unit 11 is given in the section under PREGA software organization.

LANGUAGE

Tne program is written in DEC RT-11 FORTRAN

INPUTS

Keyboard — Logical unit 5, all operator interaction and input is from keyboard

Logical Unit 11 — For modifying an existing UEDIT output file

OUTPUTS

Logical Unit 6 — Record of operator interaction with UEDIT. This is usually defined as the keyboard unit the operator is using for input

FILE.ELEMENT — Logical unit 11 output file for UEDIT saved on disk for use with SEDIT and PREGA

A sample copy of the output from UEDIT is given in Appendix A.

VARIABLES (UEDIT)

NAME (DIMENSIONS)	USAGE
A(32)	FILE.ELEMENT location on disk of the unknown sample spectrum, background spectrum or residual spectrum
ANS	Alpha-numeric I/O variable
BA	Background spectrum disk location (FILE.ELEMENT (DEVICE))
BG	Background supplied option
BP	Pivot on background allowed option
BR	Start analysis with background only option
BS	Subtract background from sample option
BT	Counting time for background in seconds
CC	Switch program to change parameter mode
CRT	In logical unit number = 5
CT	Counting time of sample in seconds
DT	Decay time of sample, set = 1
EDT	Program pointer
FØ	Rejection coefficient for partial F test
FNUM	R*4 I/O variable
GN	Gain shift ratio no longer used, set = 1
I	Index for loop over location of spectra
II	Index for program control
ITT	Upper bound of loop for output
IVAR	Record number for direct-access I/O to logical unit 11 file
K	Program control variable
LENGTH	Used as check for carriage return to exit current mode of program or exit program
LL	Switch program to list mode

VARIABLES (UEDIT)

NAME (DIMENSIONS)	USAGE
LUN	Logical unit 11 for output
M	First record of LU 11 file
MF	Result multiplication factor option (usually set = 1)
NN	Switch program to create a new file mode
NO	N, the answer no to UEDIT prompt question
NS	Number of standards from library to be used in least-squares fit by PREGA
NUM	I*2 I/O variable
NUM1	Control variable for record number in LU-11 file
OR	Number of library standard in order of desired printout
RC	Rejection coefficient for rejection of standards used. No longer used.
RO	Output residuals option
RS	Residual spectrum disk location (FILE.ELEMENT (,DEVICE))
SB	Subtract background from standards option
TH	Threshold shift, no longer used. Set = 1
UN	Unknown spectrum disk location (FILE.ELEMENT (,DEVICE))
VR	Volume reduction factor option, usually set = 1
WD	Weighting factor determination option
WF	Weighting factor calculation option
YES	Y, the answer to a UEDIT prompt question

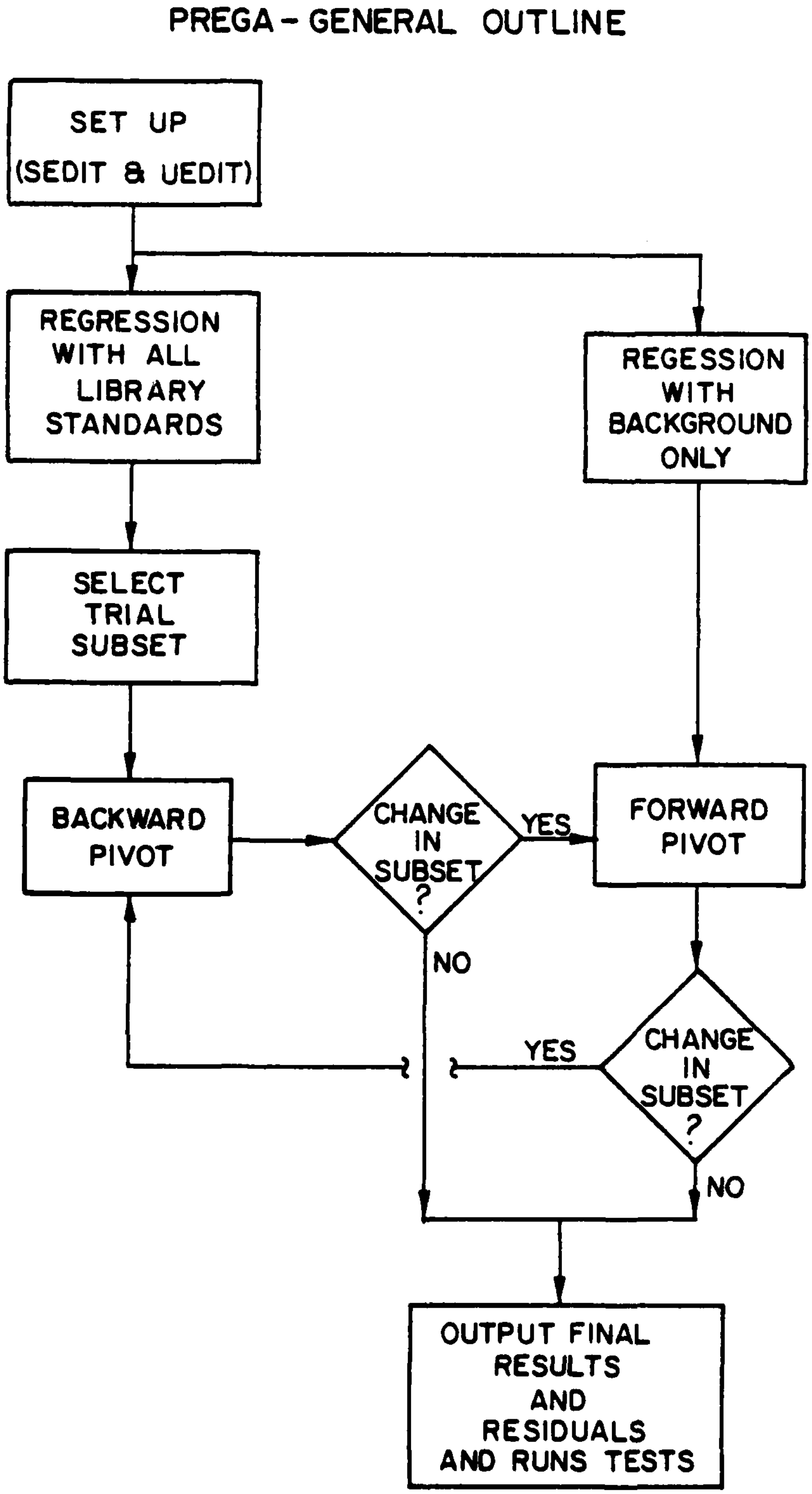


Figure 1. Summary flow chart of the program PREGA showing the major sections of the code.

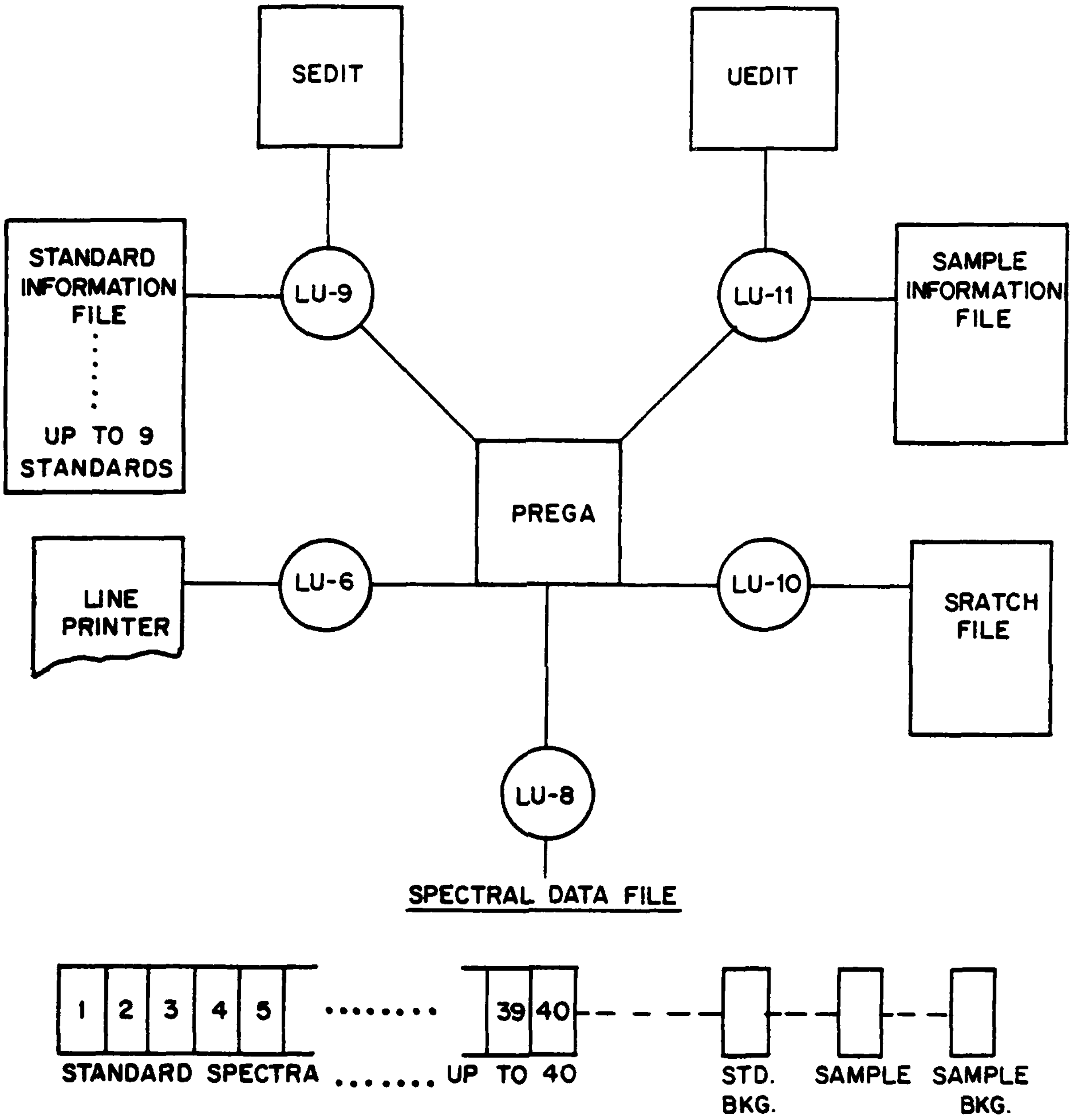


Figure 2. Logical unit and disk file structure for the program PREGA.

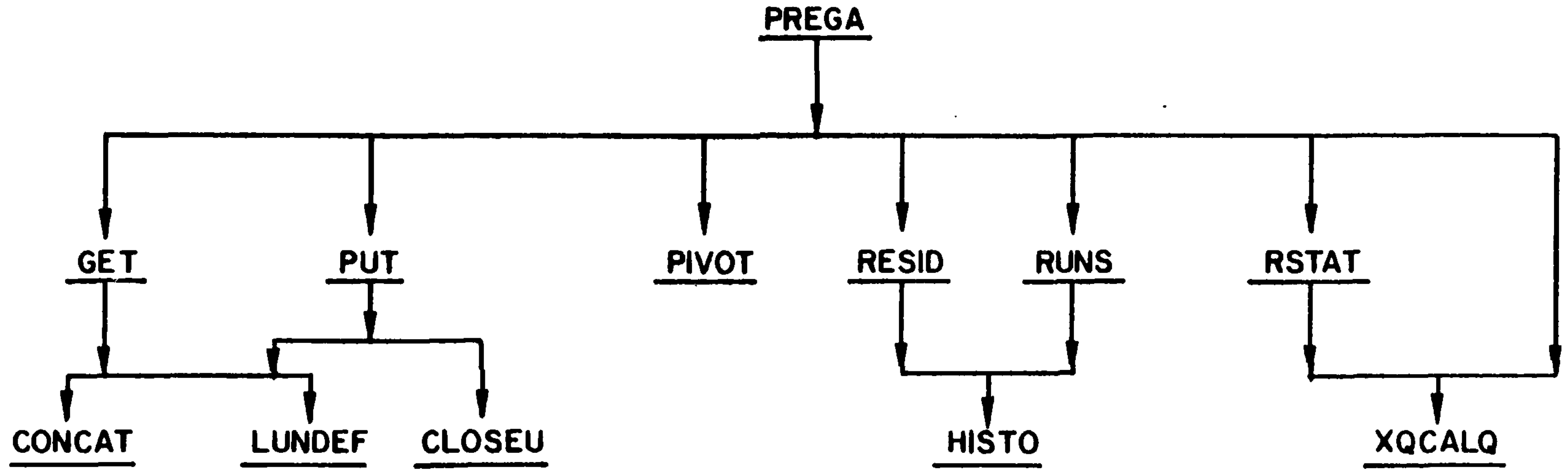


Figure 3. Hierarchy of subroutines called by the program PREGA.

REFERENCES

1. G.W. Phillips and K.W. Marlow, "Program HYPERMET for Automatic Analysis of Gamma-Ray Spectra from Germanium Detectors," NRL Memorandum Report No. 3198, January 1976.

2. E. Schonfeld, "Alpha M - An Improved Computer Program for Determining Radioisotopes by Least-Squares Resolution of the Gamma-Ray Spectra," ORNL-3975, September 1966.

3. M.J. Garside, Appl. Stat. 20, 111 (1971).

4. N.R. Draper and H. Smith, Applied Regression Analysis, 2nd. ed. (John Wiley and Sons, New York 1981).

5. ND6600 NAI Data Reduction Package Operational Instruction Documentation (Nuclear Data, Inc. publication, March 1978).

6. K.V. Mardia, J.T. Kent, and J.M. Bibby, "Multivariate Analysis," Academic Press, London (1979).

7. T. Gosnell, Lawrence Livermore National Laboratory, Livermore, CA, private communication.

APPENDIX A: Sample Output for PREGA

Sample Output for Least Squares Analysis

The following pages contain sample outputs from the editor programs UEDIT and SEDIT, and from the program PREGA. The listing from UEDIT and SEDIT are provided to show the various options used in PREGA. The least squares analysis method used is the full PREGA mode. This mode uses a trial set of standards comprised of those standards which are most highly correlated with the unknown spectrum, as determined from a fit to the complete library.

```
        PREGA UNKNOWN SAMPLE PARAMETERS

(UN) UNKNOWN SPECTRUM: FILE.ELEMENT(,DEVICE)

     : PREGA.G12161

(GN) GAIN SHIFT RATIO :     1.00000

(TH) THRESHOLD SHIFT :     1.00000

(BG) BACKGROUND SUPPLIED ? : Y

(BS) SUBTRACT BACKGROUND ? : Y

(BT) COUNTING TIME FOR BACKGROUND (SECS) :    380.000

(CT) COUNTING TIME FOR UNKNOWN SAMPLE :    430.000

(VR) VOLUME REDUCTION (OR INVERSE OF DILUTION)
     FACTOR :     1.00000

(DT) DECAY TIME :   0.000000

(MF) RESULT MULTIPLICATION FACTOR :     1.00000

(BA) BACKGROUND SPECTRUM: FILE.ELEMENT(,DEVICE)

     : PREGA.B12161

(WD) HOW ARE WEIGHTING FACTORS TO BE DETERMINED ?
     0: BASED ON ACTUAL COUNTS/CHANNEL.
     1: BASED ON CALCULATED COUNTS/CHN :     0

(NS) NUMBER OF ISOTOPES USED FROM THE STANDARD LIBRARY?:      6

(F0) F-TEST REJECTION COEFFICIENT.:    2.50000

(WF) WEIGHTING FACTOR CALCULATION.
     -1: (COUNTS)**-1
      0: (SIGMA)**-2
      1: 1 :     0

(RC) REJECTION COEFICIENT APPLIED ? : Y

(BR) BACKGROUND REGRESSION ?
      0: TRIAL SET = CHOSEN STANDARDS
      1: TRIAL SET = BACKGROUND ONLY:     1

(BP) PIVOT ON BACKGROUND SPECTRUM ALLOWED?
      0: NO PIVOT ON BACKGROUND ALLOWED
      1: ALLOW PIVOT ON BACKGROUND SPECTRUM:     0

(RO) OUTPUT RESIDUALS ?: N

(RS) RESIDUAL SPECTRUM: FILE.ELEMENT(,DEVICE)
     : DDATB.R31020

(OR) NUMBER OF LIBRARY STANDARD IN ORDER OF DESIRED PRINTOUT
     :   1  2  3  4  5  7
```

```
            PREGA LIBRARY HEADER

(NC) NUMBER OF CHANNELS PER STANDARD (MAX-512) :  255

(IT) NUMBER OF ITERATIONS FOR THRESHOLD & GAIN CALC. :     1

(BS) BACKGROUND SUBTRACT ? : N

(IS) INITIAL CHANNEL FOR COMPUTATION :    19

(IE) FINAL CHANNEL FOR COMPUTATION :  242

(BT) COUNTING TIME OF BACKGROUND (SECS.) :  380.000

(RC) REJECTION COEFICIENT :  1.00000

(BA) BACKGROUND SPECTRUM (FILE.ELEMENT(,DEVICE))

     : PREGA.B12161
```

```
            PREGA LIBRARY STANDARDS USED

STANDARD *  1 : B12161
FILE.ELEMENT(,DEVICE) : PREGA.B12161
HALF-LIFE (SECS):  1.0000E 09
COUNTING TIME OF STANDARD (SECS):  3.8000E 02
ACTIVITY:  1.0000E 00


STANDARD *  2 : 226RA
FILE.ELEMENT(,DEVICE) : NAI.RAD45
HALF-LIFE (SECS):  1.0000E 09
COUNTING TIME OF STANDARD (SECS):  1.4800E 03
ACTIVITY:  1.0000E 00


STANDARD *  3 : TH232
FILE.ELEMENT(,DEVICE) : NAI.TH232
HALF-LIFE (SECS):  1.0000E 09
COUNTING TIME OF STANDARD (SECS):  1.0000E 03
ACTIVITY:  1.0000E 00


STANDARD *  4 : K40
FILE.ELEMENT(,DEVICE) : NAI.K40
HALF-LIFE (SECS):  1.0000E 09
COUNTING TIME OF STANDARD (SECS):  1.0000E 03
ACTIVITY:  1.0000E 00


STANDARD *  5 : PCO60
FILE.ELEMENT(,DEVICE) : NAI.PCO60
HALF-LIFE (SECS):  1.0000E 09
COUNTING TIME OF STANDARD (SECS):  3.8000E 02
ACTIVITY:  1.0000E 00


STANDARD *  7 : CS137
FILE.ELEMENT(,DEVICE) : NAI.CS137
HALF-LIFE (SECS):  1.0000E 09
COUNTING TIME OF STANDARD (SECS):  1.0000E 03
ACTIVITY:  1.0000E 00
```

OUTPUT FOR PREGA ANALYSIS PROGRAM

```
SAMPLE SPECTRUM: PREGA.G12161
         0.         0.         0.         0.        -1.         1.       153.
      1620.      2227.      2427.      2644.      2579.      2669.      2451.
      2238.      2447.      2079.      2254.      1918.      2065.      1680.
      2007.      1396.      1423.      1278.      1179.      1397.      1145.
      1062.       847.       370.       775.       959.       671.       547.
       509.       667.       601.       660.       611.       517.       477.
       472.       516.       311.       445.       520.       407.       344.
       360.       335.       425.       388.       371.       277.       412.
       313.       398.       339.       352.       404.       300.       356.
       299.       268.       284.       336.       223.       292.       290.
       337.       250.       265.       230.       276.       200.       252.
       305.       213.       236.       166.       134.       269.       170.
       126.       168.       249.       197.       203.       266.       123.
       262.       216.       241.       210.       306.       358.       371.
       376.       348.       280.       310.       233.       204.       168.
       158.       170.       231.       242.       266.       232.       273.
       243.       269.       139.       140.       159.        68.        76.
       -26.        41.        44.       -14.       -36.       -15.       -43.
       -48.       -64.       -23.         3.        -3.       -22.         2.
       -15.       -28.        -9.       -17.        13.       -35.       -14.
       -23.        -7.         9.        -7.        28.         9.       -27.
       -16.       -10.        14.       -28.         9.       -16.       -22.
       -21.         2.        24.        -9.       -16.        -4.         3.
       -18.         4.       -14.         2.        -2.        15.         3.
        -4.        14.        -0.       -17.       -16.        -4.         5.
        10.       -10.       -13.        11.        21.       -30.         7.
        -0.        -4.       -17.        -3.         4.        29.        -7.
       -20.        -4.        16.        -0.         7.        -4.        22.
       -13.         9.         2.         8.         3.        13.        -7.
         7.         7.         2.        -3.        12.         4.        -7.
        14.        -2.        17.       -17.       -13.         4.         4.
       -10.       -24.       -23.       -24.         6.        -0.        -3.
        -5.         1.         8.         1.        -4.       -18.        -3.
       -15.         4.         2.        -0.         7.        -5.        -8.
         0.        -8.        -2.        -9.         2.        -2.         2.
        -6.        -3.        -5.         0.        -2.        -6.        -2.
        -7.        -0.        -1.
```

```
BACKGD SUM=   1.732E 05     SAMPLE SUM=   4.606E 04

***** CORRELATION MATRIX *****
                 B12161    226RA     TH232     K40       PC060     CS137     G12161
 B12161          1.0000    0.9533    0.8898    0.4003    0.8529    0.5202    0.8449
 226RA                     1.0000    0.7825    0.2955    0.8308    0.5252    0.8462
 TH232                               1.0000    0.3491    0.7614    0.4955    0.7405
 K40                                           1.0000    0.3640    0.1447    0.3216
 PC060                                                   1.0000    0.4746    0.9555
 CS137                                                             1.0000    0.4954
 G12161                                                                      1.0000

 WEIGHTED SUMS OF STANDARDS AND UNKNOWN
 8.7470E 01 9.4040E 01 7.0532E 02 4.3630E 01
 9.4290E 01 1.0245E 02 0.0000E-01
```

```
RESULTS OF FULL SET REGRESSION ON G12161


DEGREES OF FREEDOM=       218.
             RSS=  4.513E 02
             MSR=  2.070E 00


  B12161     -4.3499E-02  +-    2.85E-02   F=  2.3347E 00   Q=  9.3704E-01   X= -1.5303E 00   RMSR=  2.0828E 00
  226RA       1.6320E-01  +-    4.56E-02   F=  1.2789E 01   Q=  1.4521E-04   X=  3.6256E 00   RMSR=  2.1816E 00
  TH232       2.0187E-03  +-    5.30E-03   F=  1.4523E-01   Q=  3.5171E-01   X=  3.8072E-01   RMSR=  2.0621E 00
  K40        -8.9049E-03  +-    1.64E-02   F=  2.9365E-01   Q=  7.0589E-01   X= -5.4146E-01   RMSR=  2.0635E 00
  PC060       1.2711E-01  +-    5.47E-03   F=  5.3934E 02   Q=  0.0000E-01   X=  9.9999E 09   RMSR=  7.1591E 00
  CS137       3.7913E-03  +-    3.01E-03   F=  1.5861E 00   Q=  1.0378E-01   X=  1.2603E 00   RMSR=  2.0757E 00


RESULTS OF TEST SET REGRESSION ON G12161


DEGREES OF FREEDOM=       220.
             RSS=  4.524E 02
             MSR=  2.057E 00


  B12161     -3.9852E-02  +-    1.77E-02   F=  5.0691E 00   Q=  9.8814E-01   X= -2.2620E 00   RMSR=  2.0944E 00
  226RA       1.6175E-01  +-    3.68E-02   F=  1.9334E 01   Q=  3.5809E-06   X=  4.4918E 00   RMSR=  2.2271E 00
  PC060       1.2698E-01  +-    5.41E-03   F=  5.5020E 02   Q=  0.0000E-01   X=  9.9999E 09   RMSR=  7.1671E 00
  CS137       4.0606E-03  +-    2.96E-03   F=  1.8771E 00   Q=  8.5109E-02   X=  1.3714E 00   RMSR=  2.0647E 00
```

```
RESULTS OF BACKWARD REGRESSION ON G12161


DEGREES OF FREEDOM=       221.
             RSS=  4.563E 02
             MSR=  2.065E 00


  B12161     -3.8629E-02  +-    1.77E-02   F=  4.7560E 00   Q=  9.8573E-01   X= -2.1902E 00   RMSR=  2.0996E 00
  226RA       1.6712E-01  +-    3.66E-02   F=  2.0792E 01   Q=  1.5617E-06   X=  4.6658E 00   RMSR=  2.2487E 00
  PC060       1.2740E-01  +-    5.42E-03   F=  5.5352E 02   Q=  0.0000E-01   X=  9.9999E 09   RMSR=  7.2032E 00


CALCULATED PARTIAL F-,Q- AND X-VALUES  TO ADD:
TH232       F=  4.9604E-01   Q=  2.4075E-01   X=  7.0390E-01   RHO=  4.7431E-02   AMSR=  2.0694E 00
K40         F=  4.9536E-01   Q=  7.5909E-01   X= -7.0342E-01   RHO= -4.7398E-02   AMSR=  2.0694E 00
```

```
RESULTS OF BACKWARD REGRESSION ON G12161


DEGREES OF FREEDOM=       221.
             RSS=  4.563E 02
             MSR=  2.065E 00


  B12161     -3.8629E-02  +-    1.77E-02   F=  4.7560E 00   Q=  9.8573E-01   X= -2.1902E 00   RMSR=  2.0996E 00
  226RA       1.6712E-01  +-    3.66E-02   F=  2.0792E 01   Q=  1.5617E-06   X=  4.6658E 00   RMSR=  2.2487E 00
  PC060       1.2740E-01  +-    5.42E-03   F=  5.5352E 02   Q=  0.0000E-01   X=  9.9999E 09   RMSR=  7.2032E 00


CALCULATED PARTIAL F-,Q- AND X-VALUES  TO ADD:
TH232       F=  4.9604E-01   Q=  2.4075E-01   X=  7.0390E-01   RHO=  4.7431E-02   AMSR=  2.0694E 00
K40         F=  4.9536E-01   Q=  7.5909E-01   X= -7.0342E-01   RHO= -4.7398E-02   AMSR=  2.0694E 00
CS137       F=  1.8771E 00   Q=  8.5109E-02   X=  1.3714E 00   RHO=  9.1979E-02   AMSR=  2.0565E 00
```

```
*********************************************************
           FINAL RESULT FOR G12161
********************************************************


 FIT=  2.073E 00

RESULTS=CONCENTRATIONS AND EST STANDARD ERRORS

  B12161    -3.8629E-02  +-   1.77E-02   F=  4.7560E 00   Q=  9.8573E-01   X= -2.1902E 00   RMSR=  2.0996E 00
  226RA      1.6712E-01  +-   3.66E-02   F=  2.0792E 01   Q=  1.5617E-06   X=  4.6658E 00   RMSR=  2.2487E 00
  PC060      1.2740E-01  +-   5.42E-03   F=  5.5352E 02   Q=  0.0000E-01   X=  9.9999E 09   RMSR=  7.2032E 00

             ANALYSIS OF VARIANCE
SOURCE            DF       SS          MS          F
REGRESN.           3.   5.510E 03   1.837E 03   8.896E 02
RESIDUAL         221.   4.563E 02   2.065E 00
TOTAL            224.   5.966E 03



***** LLNL R-STATISTIC  *****

AUTO-CORRELATION COEFFICIENT.................  9.6947E-01
EXPECTATION VALUE............................ -1.3393E-02
STANDARD DEVIATION...........................  6.7722E-02
EQUIVALENT NORMAL STANDARD DEVIATIONS........  1.4513E 01
R-STAT FAP...................................  0.0000E-01




FINAL STATISTICS FOR ELIMINATED STANDARDS
TH232       F=  4.9604E-01   Q=  2.4075E-01   X=  7.0390E-01   RHO=  4.7431E-02   AMSR=  2.0694E 00
K40         F=  4.9536E-01   Q=  7.5910E-01   X= -7.0342E-01   RHO= -4.7398E-02   AMSR=  2.0694E 00
CS137       F=  1.8771E 00   Q=  8.5110E-02   X=  1.3714E 00   RHO=  9.1979E-02   AMSR=  2.0565E 00
```

```
 RATIO OF RESIDUALS OVER STD DEV PER CHANNEL FOR G12161

    0.0      0.0      0.0      0.0      0.0      0.0      0.0
    0.0      0.0      0.0      0.0      0.0      0.0      0.0
    0.0      0.0      0.0      0.0     -0.1      1.3     -0.4
    2.5     -1.1      0.4      0.0     -0.2      2.2      0.3
    0.1     -1.2      0.2     -1.9      0.8     -1.4     -2.4
   -2.3      0.7      0.7      1.8      0.7     -0.2     -0.0
    0.5      1.7     -1.2      1.1      2.1      1.0     -0.5
   -0.1     -0.6     -0.0     -0.8     -1.1     -2.6      0.3
   -0.8      1.9      0.7      1.3      2.2      0.5      1.7
    0.6     -0.1      0.3      1.2     -1.2      0.4      0.7
    1.9      0.2      0.5     -0.4      0.7     -1.1      0.4
    1.6     -0.2      0.2     -1.2     -1.6      1.8     -0.4
   -1.4     -0.0      2.1      0.7      1.2      2.8     -1.2
    2.5      1.2      2.1      0.9      3.3      4.1      3.4
    2.0     -0.6     -3.1     -1.7     -3.0     -2.0     -1.3
    0.5      2.1      4.8      4.9      5.2      2.3      1.8
   -0.8     -0.9     -5.2     -4.2     -2.0     -3.0     -1.1
   -2.7      0.3      0.9     -0.6     -0.9     -0.3     -1.4
   -1.4     -2.0     -0.7      0.4      0.2     -0.8      0.3
   -0.7     -1.5     -0.5     -1.0      0.5     -2.1     -1.3
   -1.5     -1.0     -0.4     -1.2      0.6     -0.3     -2.0
   -1.4     -1.0      0.2     -1.9      0.1     -1.3     -1.4
   -1.6     -0.1      1.8     -0.6     -1.2     -0.3      0.3
   -1.1      0.5     -0.8      0.1     -0.2      1.2      0.3
   -0.2      1.2      0.1     -0.9     -1.1     -0.3      0.3
    0.7     -0.6     -0.8      0.7      1.5     -1.7      0.3
   -0.1     -0.2     -1.1     -0.3      0.4      2.3     -0.4
   -1.2     -0.4      1.4      0.1      0.7     -0.2      2.2
   -1.1      0.7      0.2      0.8      0.3      1.2     -0.5
    0.7      0.7      0.2     -0.2      1.0      0.4     -0.2
    1.2      0.0      1.2     -0.8     -0.5      0.4      0.5
   -0.3     -1.1     -1.3     -1.5      0.7      0.1     -0.1
   -0.4      0.1      1.2      0.2     -0.4     -1.9     -0.3
   -1.6      0.7      0.4      0.0      1.0     -0.7     -0.9
    0.2     -1.0     -0.3     -1.1

SUSPICIOUS CHANNELS

 97    4.10 108    4.79 109    4.91 110    5.24

MEAN RESIDUAL =  -0.004, VARIANCE =   2.045, SKEWNESS =  0.376   , EXCESS =   1.87
```

```
      DISTRIBUTION OF RESIDUALS

   SCALE FACTOR=          1

     -5.00                1  *
     -4.50                0  :
     -4.00                1  *
     -3.50                0  :
     -3.00                3  ***
     -2.50                4  ****
     -2.00                8  ********
     -1.50               18  ******************
     -1.00               33  *********************************
    -0.500               28  ****************************
     0.000               37  *************************************
     0.500               42  ******************************************
      1.00               17  *****************
      1.50                7  *******
      2.00               14  **************
      2.50                4  ****
      3.00                1  *
      3.50                2  **
      4.00                1  *
      4.50                0  :
      5.00                3  ***

      TOTAL =          224

        DISTRIBUTION OF RUNS

   SCALE FACTOR=          1

     -10.0                0  :
     -9.00                0  :
     -8.00                1  *
     -7.00                2  **
     -6.00                2  **
     -5.00                0  :
     -4.00                6  ******
     -3.00                6  ******
     -2.00                6  ******
     -1.00               26  **************************
     0.000                0  :
      1.00               20  ********************
      2.00               14  **************
      3.00                6  ******
      4.00                3  ***
      5.00                2  **
      6.00                0  :
      7.00                2  **
      8.00                1  *
      9.00                0  :
      10.0                0  :

      TOTAL =           97

   LOCATIONS AND SIZES OF LARGE RUNS

          49     -7        58     7        69     5        92     8       100     -6
         106      7       113    -8       123    -7       139    -6       198      5
```

Appendix B: FORTRAN Listings, DEC RT-11

1. PKANAL
2. PEAKNL
3. FREEFM
4. GSHIFT
5. REGAIN
6. RECHAN
7. SEDIT
8. UEDIT
9. PREGA
10. GET
11. HISTO
12. PIVOT
13. PUT
14. RESID
15. RSTAT
16. RUNS
17. XQCALC

```
MIDAS FORTRAN IV          1 JAN 1981 12:23:03 AM      PAGE 001

      C     PROGRAM PKANAL
      C     WRITTEN BY G.W.PHILLIPS, JULY 1981
      C
      C     READS IN DATA FROM NUCLEAR DATA SPECTRAL FILES
      C     AND CALCULATES PEAK POSITIONS, WIDTHS AND AREAS
      C     FOR SELECTED REGIONS
      C
      C     INPUT: JL,JR - LOWER AND UPPER CHANNELS FOR PEAK REGIONS
      C     ASC - FILENAME FOR SPECTRUM TO BE ANALYZED
      C
      C     OUTPUT: JL,JR - INPUT LIMITS FOR PEAK REGIONS
      C     KL,KR - PEAK LIMITS (POSITIVE NET COUNT)
      C     AMAX - PEAK MAXIMUM (NET COUNTS)
      C     AREA - PEAK NET AREA
      C     BKGD - BACKGROUND AREA BENEATH PEAK
      C     CG - PEAK CENTROID
      C     VAR - PEAK VARIANCE ABOUT CENTROID
      C     FWHM - PEAK FULL WIDTH AT HALF MAXIMUM
      C     ENERGY - CALIBRATED PEAK ENERGY
      C
      C     PROCEDURES CALLED:
      C     PEAKNL - DOES PEAK ANALYSIS
      C     GET - READS SPECTRUM AND ENERGY CALIBRATION FORD DISK
      C
```

```
C
      COMMON/ARRAY/ARRAY(512)
      INTEGER CRT,DATA,PERIOD,BLANK,JL(20),JR(20)
      COMMON/DATA/DATA(80)/FREE/INTEG(16),REALX(16),ALPHA(16)
      REAL*8 ALPHA,ABLANK,AHEAD(11)
      LOGICAL*1 ASC(60)
      DATA ABLANK/8H        /
      DATA CRT/5/,LP/6/,IN/5/,IOUT/6/,PERIOD/1H./,BLANK/1H /,LUF/8/
      DATA AHEAD/8H REGION ,8HLIMITS  ,8H   PEAK ,8HLIMITS  ,
     1            8H HEIGHT ,8H  AREA  ,8H  BKGD. ,
     2            8HCENTROID,8HVARIANCE,8H  FWHM  ,
     3            8H ENERGY /
C
100   WRITE(CRT,110)
110   FORMAT(1X,'ENTER REGIONS BY FIRST,LAST CHANNEL'/)
      DO 160 I=1,20
      READ(CRT,120)LEN,DATA
120   FORMAT(Q,80A1)
      IF(LEN.LT.1) GOTO 180
      NX=2
      MX=1
      NA=1
      CALL FREEFM(NX,MX,NA,1)
      JL(I)=INTEG(1)
      JR(I)=INTEG(2)
160   CONTINUE
      I=21
180   N=I-1
      IF(N.LT.1) GOTO 990
C
```

```
C
C
200     IFORM=1
        WRITE(CRT,210)
210     FORMAT(1H0,'FILENAME FOR SPECTRUM'/)
220     READ(CRT,230)LEN,ASC
230     FORMAT(Q,80A1)
        IF(LEN.LT.1) GOTO 990
        WRITE(IOUT,240) IFORM,(ASC(I),I=1,LEN)
240     FORMAT(I1,80A1)
        IFORM=0
        WRITE(IOUT,250) AHEAD
250     FORMAT(2X,4A8,1X,A8,6(4X,A8))
        MI=256
        CALL GET(ASC,ARRAY,IVAR,LUF,MI,A0,B0,C0)
        WRITE(CRT,260) MI
260     FORMAT(1X,I5,' CHANNELS READ IN'/)
C
C
300     DO 360 I=1,N
        KL=JL(I)
        KR=JR(I)
        CALL PEAKNL(KL,KR,ML,MR,MX,XL,XR,
     1              AL,BL,AMAX,AREA,BKGD,CG,VAR,FWHM)
        ENERGY=A0+CG*(B0+CG*C0)
        WRITE(IOUT,320)JL(I),JR(I),KL,KR,AMAX,AREA,BKGD,
     1              CG,VAR,FWHM,ENERGY
320     FORMAT(2(I8,I6,2X),1X,7G12.4)
360     CONTINUE
        GOTO 220
C
990     WRITE(IOUT,991)
991     FORMAT(1H1)
        STOP
        END
```

```
MIDAS FORTRAN IV   STORAGE MAP


NAME    OFFSET  ATTRIBUTES

JL      000006  INTEGER*2 ARRAY (20)
JR      000056  INTEGER*2 ARRAY (20)
AHEAD   000126  REAL*8    ARRAY (11)
ASC     000256  LOGICAL*1 ARRAY (60)
CRT     000362  INTEGER*2 VARIABLE
PERIOD  000372  INTEGER*2 VARIABLE
BLANK   000374  INTEGER*2 VARIABLE
ABLANK  000352  REAL*8    VARIABLE
LP      000364  INTEGER*2 VARIABLE
IN      000366  INTEGER*2 VARIABLE
IOUT    000370  INTEGER*2 VARIABLE
LUF     000376  INTEGER*2 VARIABLE
I       000630  INTEGER*2 VARIABLE
LEN     000632  INTEGER*2 VARIABLE
NX      000634  INTEGER*2 VARIABLE
MX      000636  INTEGER*2 VARIABLE
NA      000640  INTEGER*2 VARIABLE
FREEFM  000000  REAL*4    PROCEDURE
N       000642  INTEGER*2 VARIABLE
IFORM   000644  INTEGER*2 VARIABLE
MI      000646  INTEGER*2 VARIABLE
GET     000000  REAL*4    PROCEDURE
IVAR    000650  INTEGER*2 VARIABLE
A0      000652  REAL*4    VARIABLE
B0      000656  REAL*4    VARIABLE
C0      000662  REAL*4    VARIABLE
KL      000666  INTEGER*2 VARIABLE
KR      000670  INTEGER*2 VARIABLE
PEAKNL  000000  REAL*4    PROCEDURE
ML      000672  INTEGER*2 VARIABLE
MR      000674  INTEGER*2 VARIABLE
XL      000676  REAL*4    VARIABLE
XR      000702  REAL*4    VARIABLE
AL      000706  REAL*4    VARIABLE
BL      000712  REAL*4    VARIABLE
AMAX    000716  REAL*4    VARIABLE
AREA    000722  REAL*4    VARIABLE
BKGD    000726  REAL*4    VARIABLE
CG      000732  REAL*4    VARIABLE
VAR     000736  REAL*4    VARIABLE
FWHM    000742  REAL*4    VARIABLE
ENERGY  000746  REAL*4    VARIABLE

COMMON BLOCK /ARRAY/    LENGTH 004000

ARRAY   000000  REAL*4    ARRAY (512)

COMMON BLOCK /DATA/     LENGTH 000240

DATA    000000  INTEGER*2 ARRAY (80)

COMMON BLOCK /FREE/     LENGTH 000340

INTEG   000000  INTEGER*2 ARRAY (16)
REALX   000040  REAL*4    ARRAY (16)
ALPHA   000140  REAL*8    ARRAY (16)
```

```
MIDAS FORTRAN IV          11 AUG 1983  5:19:01 PM      PAGE 001

0001        SUBROUTINE PEAKNL(KL,KR,ML,MR,MX,XL,XR,
           1            AL,BL,AMAX,AREA,BKGD,CG,VAR,FWHM)
      C     WRITTEN BY G.W. PHILLIPS, JULY 1981
      C
      C     FINDS THE AREA, CENTER OF GRAVITY, VARIANCE AND FWHM
      C     FOR A PEAK IN A REGION BETWEEN KL AND KR
      C     AFTER SUBTRACTING A BACKGROUND CALCULATED
      C     AS A STRAIGHT LINE DRAWN BETWEEN
      C     THE MINIMUM BELOW THE PEAK AND THE MINIMUM
      C     ABOVE THE PEAK.
      C     IF THE DATA IS FLAT WITHIN STATISTICS AT THE MINIMUM
      C     THE AVERAGE OF THE THREE CHANNELS
      C     INCLUDING THE MINIMUM IS USED.
      C
      C     INPUT: KL,KR - INITIAL LIMITS FOR PEAK REGION
      C
      C     OUTPUT: KL,KR - FINAL LIMITS FOR PEAK (POSITIVE NET COUNT)
      C     ML,MR - MINIMUM CHANNELS BELOW AND ABOVE PEAK
      C     MX - PEAK MAXIMUM CHANNEL
      C     XL,XR - HALF MAXIMUM POINTS BELOW AND ABOVE PEAK
      C     AL,BL - SLOPE AND OFFSET (FROM ML)
      C            FOR BACKGROUND
      C     AMAX - PEAK MAXIMUM (NET COUNTS)
      C     AREA - PEAK NET AREA
      C     BKGD - BACKGROUND AREA BENEATH PEAK
      C     CG - PEAK CENTROID
      C     VAR - PEAK VARIANCE ABOUT CENTROID
      C     FWHM - PEAK FULL WIDTH AT HALF MAXIMUM
```

```
C
              COMMON/ARRAY/AX(512)
D             DATA ICRT/5/
C
C             LOOK FOR MAXIMUM IN THE REGION
C
      100     AMAX=0.
              MX=0
              DO 110 I=KL,KR
              IF(AMAX.GT.AX(I)) GOTO 110
              AMAX=AX(I)
              MX=I
      110     CONTINUE
C
C             LOOK FOR LOWER AND UPPER MINIMA
C
              AMINL=AX(KL)
              ML=KL
              DO 130 I=KL,MX
              IF(AX(I).GT.AMINL) GOTO 130
              AMINL=AX(I)
              ML=I
      130     CONTINUE
              AMINR=AX(MX)
              MR=MX
              DO 140 I=MX,KR
              IF(AX(I).GE.AMINR) GOTO 140
              AMINR=AX(I)
              MR=I
      140     CONTINUE
      C
      C     DETERMINE BACKGROUND PARAMETERS
      C
0027  200   IF(ML.EQ.KL) ML=ML+1
0029        J=ML-1
0030        K=ML+1
0031        SUML=0.
0032        SQL=0.
0033        DO 210 I=J,K
0034        FL=AX(I)
0035        SUML=SUML+FL
0036        SQL=SQL+FL**2
0037  210   CONTINUE
0038        IF(MR.EQ.KR) MR=MR-1
0040        J=MR-1
0041        K=MR+1
0042        SUMR=0.
0043        SQR=0.
0044        DO 220 I=J,K
0045        FR=AX(I)
0046        SUMR=SUMR+FR
0047        SQR=SQR+FR**2
0048  220   CONTINUE
0049        DL=AMAX1(ABS(AX(ML-1)-AX(ML)),ABS(AX(ML+1)-AX(ML)))
0050        DR=AMAX1(ABS(AX(MR-1)-AX(MR)),ABS(AX(MR+1)-AX(MR)))
0051        AL=SUML/3.
0052        IF(DL.LT.2.*SQL) GOTO 240
0054        IF(AX(ML).GT.AX(ML-1)) ML=ML-1
0056        AL=AX(ML)
0057  240   AR=SUMR/3.
0058        IF(DR.LT.2.*SQR) GOTO 250
0060        IF(AX(MR).GT.AX(MR+1)) MR=MR+1
0062        AR=AX(MR)
0063  250   BL=(AR-AL)/FLOAT(MR-ML)
      D     WRITE(ICRT,260)ML,MR,AL,BL
      D260  FORMAT(1X,'BACKGROUND PARAMETERS: ML MR AL BL'/
      D     1           1X,2I5,2G12.4)
      C
      C     LOOK FOR MAXIMUM IN NET COUNTS
      C
0064        MX=ML
0065        AMAX=AX(ML)-AL
0066        DO 270 I=ML,MR
0067        A=AX(I)-AL-BL*FLOAT(I-ML)
0068        IF(A.LT.AMAX) GOTO 270
0070        AMAX=A
0071        MX=I
0072  270   CONTINUE
      C
      C
      C     DEFINE PEAK LIMITS
      C
0073  300   KL=ML-1
0074        DO 310 I=ML,MX
0075        IF(AX(I).LT.AL+BL*FLOAT(I-ML)) KL=I
0077  310   CONTINUE
0078        KL=MIN0(KL+1,MX)
0079        KR=MX
0080        DO 320 I=MX,MR
0081        IF(AX(I).LT.AL+BL*FLOAT(I-ML)) GOTO 330
0083        KR=I
0084  320   CONTINUE
      C
      C     CALCULATE PEAK PARAMETERS
      C
0085  330   AI=0.
0086        AISQ=0.
0087        AREA=0.
0088        BKGD=0.
0089        CG=0.
0090        VAR=0.
0091        FWHM=0.
      D     WRITE(ICRT,340)
      D340  FORMAT(4X,'I',3X,' COUNTS ',3X,'  BKGD. ',3X,'  NET    ',
      D    1            3X,'  AREA   ',3X,'   AI    ',3X,'  AISQ  ')
0092        DO 360 I=KL,KR
0093        B=AL+BL*FLOAT(I-ML)
0094        BKGD=BKGD+B
0095        A=AX(I)-B
0096        AREA=AREA+A
0097        AI=AI+A*FLOAT(I)
0098        AISQ=AISQ+A*FLOAT(I)**2
      D     WRITE(ICRT,350)I,AX(I),B,A,AREA,AI,AISQ
      D350  FORMAT(1X,I4,6G11.3)
0099  360   CONTINUE
0100        IF(AREA.EQ.0.) RETURN
0102        CG=AI/AREA
0103        VAR=(AISQ+AREA*CG**2-2.*AI*CG)/AREA
```

```
C
C     FIND FWHM
C
400   AHALF=AMAX/2.
      B=AL+BL*FLOAT(KL-ML-1)
      AO=AX(KL-1)-B
      DO 410 I=KL,MX
      IL=I
      B=B+BL
      A=AX(I)-B
      IF(A.GT.AHALF) GOTO 420
      AO=A
410   CONTINUE
420   DA=A-AO
      IF(DA.EQ.0.) DA=2.*(A-AHALF)
      X=(A-AHALF)/DA
      XL=FLOAT(IL)-X
C
      B=AL+BL*FLOAT(MX-ML-1)
      AO=AX(MX-1)-B
      DO 430 I=MX,KR
      IR=I
      B=B+BL
      A=AX(I)-B
      IF(A.LT.AHALF) GOTO 440
      AO=A
430   CONTINUE
      IR=IR+1
440   DA=AO-A
      IF(DA.EQ.0.) DA=2.*(AHALF-A)
      X=(AHALF-A)/DA
      XR=FLOAT(IR)-X
      FWHM=XR-XL
D     WRITE(ICRT,450) MX,IL,IR,XL,XR,FWHM
D450  FORMAT(1X,'WIDTH PARAMETERS: MX IL IR XL XR FWHM'/
D    1            1X,3I8,3G12.4/)
C
      RETURN
      END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

KL       000014  INTEGER*2 PARAMETER VARIABLE
KR       000016  INTEGER*2 PARAMETER VARIABLE
ML       000020  INTEGER*2 PARAMETER VARIABLE
MR       000022  INTEGER*2 PARAMETER VARIABLE
MX       000024  INTEGER*2 PARAMETER VARIABLE
XL       000026  REAL*4    PARAMETER VARIABLE
XR       000030  REAL*4    PARAMETER VARIABLE
AL       000032  REAL*4    PARAMETER VARIABLE
BL       000034  REAL*4    PARAMETER VARIABLE
AMAX     000036  REAL*4    PARAMETER VARIABLE
AREA     000040  REAL*4    PARAMETER VARIABLE
BKGD     000042  REAL*4    PARAMETER VARIABLE
CG       000044  REAL*4    PARAMETER VARIABLE
VAR      000046  REAL*4    PARAMETER VARIABLE
FWHM     000050  REAL*4    PARAMETER VARIABLE
I        000052  INTEGER*2 VARIABLE
AMINL    000054  REAL*4    VARIABLE
AMINR    000060  REAL*4    VARIABLE
J        000064  INTEGER*2 VARIABLE
K        000066  INTEGER*2 VARIABLE
SUML     000070  REAL*4    VARIABLE
SQL      000074  REAL*4    VARIABLE
FL       000100  REAL*4    VARIABLE
SUMR     000104  REAL*4    VARIABLE
SQR      000110  REAL*4    VARIABLE
FR       000114  REAL*4    VARIABLE
DL       000120  REAL*4    VARIABLE
AMAX1    000000  REAL*4    PROCEDURE
ABS      000000  REAL*4    PROCEDURE
DR       000124  REAL*4    VARIABLE
AR       000130  REAL*4    VARIABLE
FLOAT    000000  REAL*4    PROCEDURE
A        000134  REAL*4    VARIABLE
MIN0     000000  INTEGER*2 PROCEDURE
AI       000140  REAL*4    VARIABLE
AISQ     000144  REAL*4    VARIABLE
B        000150  REAL*4    VARIABLE
AHALF    000154  REAL*4    VARIABLE
AO       000160  REAL*4    VARIABLE
IL       000164  INTEGER*2 VARIABLE
DA       000166  REAL*4    VARIABLE
X        000172  REAL*4    VARIABLE
IR       000176  INTEGER*2 VARIABLE

COMMON BLOCK /ARRAY/    LENGTH 004000
```

```
0001          SUBROUTINE FREEFM(N,M,NA,ITYPE)
      C       LAST REVISED AUGUST 1981 BY G.W.P.
      C
      C       GENERAL SUBROUTINE TO DECODE DATA READ IN FREE FIELD FORMAT
      C       DELIMITERS ARE EITHER A BLANK OR A COMMA
      C       THE ROUTINE ASSUMES THE DATA HAS BEEN READ INTO ARRAY IDATA WITH
      C          THE FORMAT (80A1)
      C       N IS THE NUMBER OF DATA ELEMENTS, MAXIMUM=16
      C       N IS RETURNED AS THE NUMBER OF DATA ELEMENTS FOUND
      C       M IS THE LOCATION IN THE ARRAY FOR STORING THE FIRST DATA ELEMENT
      C       M IS RETURNED AS THE LOCATION FOLLOWING THE NTH DATA ELEMENT
      C       NA IS THE BEGINNING COLUMN OF THE DATA
      C       NA IS RETURNED AS THE COLUMN FOLLOWING THE NTH DATA ELEMENT
      C       ITYPE IS THE TYPE OF DATA,
      C          1=INTEGER
      C          2=REAL
      C          3=ALPHANUMERIC
      C
0002          COMMON/DATA/IDATA(80)
0003          COMMON/FREE/INTEG(16),REALX(16),ALPHA(16)
0004          INTEGER SEMI,E,COMMA
0005          REAL*8 ALPHA,BLANK
0006          DIMENSION ITEMP(20),AFORM(2)
0007          DATA SEMI,E,IBLK1,COMMA,IBLK2,BLNK4,BLANK
             * /';','E',' ',',','    ','     ','        '/
      C
0008          L=M
0009          M=M+N-1
0010          DO 300 I=L,M
0011          IF(NA.GT.80) GO TO 400
      C
      C       LOOK FOR START OF CURRENT FIELD
      C
0013          DO 210 J=NA,80
0014          JQQ=J
0015             IF(IDATA(J).NE.IBLK1) GO TO 215
0017     210  CONTINUE
0018             NA=81
0019             GO TO 400
0020     215  IF(IDATA(JQQ).NE.COMMA) GO TO 220
0022             NA=JQQ+1
0023             GO TO 290
```

```
      C
      C     LOOK FOR END OF CURRENT FIELD
      C
0024   220  IL=JQQ
0025        ILQ = IL
0026   221  DO 230 J=ILQ,80
0027        JQQ=J
0028           IF(IDATA(J).EQ.IBLK1) GO TO 235
0030           IF(IDATA(J).NE.COMMA) GO TO 230
0032              IR=J-1
0033              NA=J+1
0034              GO TO 250
0035   230  CONTINUE
0036           IR=80
0037           NA=81
0038           GO TO 250
      C
      C     CHECK FOR EXPONENT
      C
0039   235  IF((ITYPE.NE.2).OR.(IDATA(JQQ-1).NE.E)) GO TO 236
0041        ILQ = JQQ + 1
0042        GO TO 221
0043   236  IR=JQQ - 1
0044        IJ=JQQ+1
      C
      C     SET NA TO START OF NEXT FIELD
      C
0045        DO 240 J=IJ,80
0046           IF(IDATA(J).EQ.IBLK1) GO TO 240
0048              NA=J
0049              IF(IDATA(J).EQ.COMMA) NA=NA+1
0051              GO TO 250
0052   240  CONTINUE
0053           NA=81
      C
      C     ENCODE DATA IN CURRENT FIELD
      C
0054   250  NI=IR-IL+1
0055        IF(NI.LT.1) GO TO 290
0057        ENCODE(NI,255,ITEMP) (IDATA(J),J=IL,IR)
0058   255  FORMAT(83A1)
```

```
      C
      C
0059          GO TO (260,270,280),ITYPE
      C
      C       DECODE INTEGER DATA
      C
0060   260    ENCODE(8,265,AFORM) NI
0061   265    FORMAT('(I'I2,')    ')
0062          DECODE(NI,AFORM,ITEMP) INTEG(I)
0063          GO TO 300
      C
      C       DECODE REAL DATA
      C
0064   270    ENCODE(8,275,AFORM) NI
0065   275    FORMAT('(E'I2,'.0) ')
0066          DECODE(NI,AFORM,ITEMP) REALX(I)
0067          GO TO 300
      C
      C       DECODE ALPHANUMERIC DATA
      C
0068   280    IF(NI.GT.8) NI=8
0070          DO 287 J=1,NI
0071          IF(ITEMP(J).EQ.SEMI) ITEMP(J)=COMMA
0073   287    CONTINUE
0074          ENCODE(8,288,AFORM)NI
0075    288   FORMAT('(A',I1,')     ')
0076          DECODE(8,AFORM,ITEMP) ALPHA(I)
0077          GO TO 300
      C
      C       BLANK OUT REMAINING DATA
      C
0078  290     INTEG(I)=IBLK2
0079          REALX(I)=BLNK4
0080          ALPHA(I)=BLANK
0081   300    CONTINUE
0082          M=M+1
0083          RETURN
      C
      C       BLANK INPUT, BLANK OUT ALL DATA
      C
0084  400     DO 410 J=I,M
0085          INTEG(J)=IBLK2
0086          REALX(J)=BLNK4
0087          ALPHA(J)=BLANK
0088  410     CONTINUE
0089          M=I
0090          N=M-L
0091          RETURN
0092          END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

ITEMP    000024  INTEGER*2 ARRAY (20)
AFORM    000074  REAL*4    ARRAY (2)
N        000014  INTEGER*2 PARAMETER VARIABLE
M        000016  INTEGER*2 PARAMETER VARIABLE
NA       000020  INTEGER*2 PARAMETER VARIABLE
ITYPE    000022  INTEGER*2 PARAMETER VARIABLE
SEMI     000104  INTEGER*2 VARIABLE
E        000106  INTEGER*2 VARIABLE
COMMA    000112  INTEGER*2 VARIABLE
BLANK    000122  REAL*8    VARIABLE
IBLK1    000110  INTEGER*2 VARIABLE
IBLK2    000114  INTEGER*2 VARIABLE
BLNK4    000116  REAL*4    VARIABLE
L        000210  INTEGER*2 VARIABLE
I        000212  INTEGER*2 VARIABLE
J        000214  INTEGER*2 VARIABLE
JQQ      000216  INTEGER*2 VARIABLE
IL       000220  INTEGER*2 VARIABLE
ILQ      000222  INTEGER*2 VARIABLE
IR       000224  INTEGER*2 VARIABLE
IJ       000226  INTEGER*2 VARIABLE
NI       000230  INTEGER*2 VARIABLE

COMMON BLOCK /DATA/      LENGTH 000240

IDATA    000000  INTEGER*2 ARRAY (80)

COMMON BLOCK /FREE/      LENGTH 000340

INTEG    000000  INTEGER*2 ARRAY (16)
REALX    000040  REAL*4    ARRAY (16)
ALPHA    000140  REAL*8    ARRAY (16)
```

```
C     PROGRAM GSHIFT
C     WRITTEN BY G.W.PHILLIPS, AUGUST 1981
C
C     READS IN DATA FROM NUCLEAR DATA SPECTRAL FILES
C     AND PERFORMS A GAIN AND ZERO OFFSET ADJUSTMENT
C
      DIMENSION S(256),R(256)
      INTEGER CRT,DATA,PERIOD,BLANK
      COMMON/DATA/DATA(80)/FREE/INTEG(16),REALX(16),ALPHA(16)
      COMMON/HEAD/IHDR,TITLE(32),ID(8),ELTIME,PLTIME,ERTIME,PRTIME
      INTEGER*2 TITLE,ID,DUMT(32),DUMID(8)
      REAL*8 ALPHA,ABLANK,AHEAD(10)
      LOGICAL*1 ASC(60),YES
      DATA CRT/5/,LP/6/,IN/5/,IOUT/6/,PERIOD/1H./,BLANK/1H /,LUF/8/
      DATA AHEAD/8H REGION ,8HLIMITS  ,8H   PEAK ,8HLIMITS  ,
     1           8H HEIGHT ,8H   AREA  ,8H  BKGD. ,
     2           8HCENTROID,8HVARIANCE,8H  FWHM  /
      DATA BLNK4/4H    /,ABLANK/8H        /,YES/1HY/
```

```
C
C
100   WRITE(CRT,110)
110   FORMAT(1H0,'FILENAME FOR SPECTRUM'/)
      READ(CRT,120)LEN,ASC
120   FORMAT(Q,80A1)
      IF(LEN.LT.1) GOTO 990
      LUF=8
      NCH=256
      CALL GET(ASC,R,IVAR,LUF,NCH,A0,B0,C0)
      WRITE(CRT,130) NCH
130   FORMAT(1X,I5,' CHANNELS READ IN'/)
D     WRITE(IOUT,140)(R(I),I=1,NCH)
D140  FORMAT(1H1,10G12.3/(1X,10G12.3))
D     WRITE(IOUT,150)
D150  FORMAT(1H1)
      WRITE(CRT,160)B0,A0,C0
160   FORMAT(1X,'ENERGY CALIBRATION FROM HEADER GIVES SLOPE,OFFSET,'
     1    ,'CURVATURE:'/3G12.4/
     2    1X,'USE THIS FOR OLD SLOPE AND OFFSET. YES OR NO?'/)
      READ(CRT,170)LEN,ASC
170   FORMAT(Q,80A1)
      IF(LEN.LT.1) GOTO 200
      IF(ASC(1).EQ.YES) GOTO 300
C
C
200   A0=0.
      B0=1.
      C0=0.
      WRITE(CRT,210)
210   FORMAT(1X,'ENTER OLD SLOPE,OFFSET'/)
      READ(CRT,220)LEN,DATA
220   FORMAT(Q,80A1)
      IF(LEN.LT.i) GOTO 300
      NX=2
      MX=1
      NA=1
      CALL FREEFM(NX,MX,NA,2)
      IF(REALX(1).NE.BLNK4) B0=REALX(1)
      IF(REALX(2).NE.BLNK4) A0=REALX(2)
  300 DO 230 I=1,32
      DUMT(I)=TITLE(I)
  230 CONTINUE
      DO 235 I=1,8
      DUMID(I)=ID(I)
  235 CONTINUE
      IDELT=ELTIME
      IDPLT=PLTIME
      IDERT=ERTIME
      IDPRT=PRTIME
      C
      C
0057        WRITE(CRT,310)
0058  310   FORMAT('0TO GET NEW SLOPE AND OFFSET FROM SPECTRAL FILE'/
           1     ' ENTER FILENAME'$)
0059        READ(CRT,320)LEN,ASC
0060  320   FORMAT(Q,80A1)
0061        IF(LEN.LT.1) GOTO 350
0063        KUF=8
0064        MCH=256
0065        CALL GET(ASC,S,JVAR,KUF,MCH,A1,B1,C1)
0066        WRITE(CRT,330)B1,A1,C1
0067  330   FORMAT(' HEADER GIVES SLOPE,OFFSET,CURVATURE:'/3F12.7/
           1     ' USE THIS FOR NEW AND OFFSET, YES OR NO?'/)
0068        READ(CRT,340) LEN,ASC
0069  340   FORMAT(Q,80A1)
0070        IF(LEN.LT.1) GOTO 350
0072        IF(ASC(1).EQ.YES) GOTO 400
      C
0074  350   A1=0.
0075        B1=1.
0076        C1=0.
0077        WRITE(CRT,355)
0078  355   FORMAT(1X,'ENTER NEW SLOPE,OFFSET'/)
0079        READ(CRT,360)LEN,DATA
0080  360   FORMAT(Q,80A1)
0081        IF(LEN.LT.1) GOTO 370
0083        NX=2
0084        MX=1
0085        NA=1
0086        CALL FREEFM(NX,MX,NA,2)
0087        IF(REALX(1).NE.BLNK4) B1=REALX(1)
0089        IF(REALX(2).NE.BLNK4) A1=REALX(2)
0091        GOTO 400
      C
0092  370   G=1.
0093        EPS=0.
0094        WRITE(CRT,380)
0095  380   FORMAT(1X,'ENTER GAIN,ZERO SHIFT'/)
0096        READ(CRT,390)LEN,DATA
0097  390   FORMAT(Q,80A1)
0098        IF(LEN.LT.1) GOTO 990
0100        NX=2
0101        MX=1
0102        NA=1
0103        CALL FREEFM(NX,MX,NA,2)
0104        IF(REALX(1).NE.BLNK4) G=REALX(1)
0106        IF(REALX(2).NE.BLNK4) EPS=REALX(2)
0108        GOTO 411
```

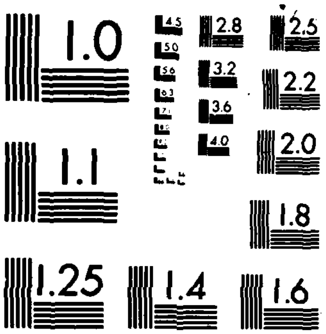
```
C
C
C     PERFORM GAIN SHIFT
C
400   G=B0/B1
      EPS=(A0-A1)/B1
      WRITE(CRT,410) G,EPS
410   FORMAT(1H0,'RESULTING GAIN AND ZERO SHIFT ARE:'/2G12.4/)
411   MF=1
      MR=NCH
      CALL REGAIN(R,S,G,EPS,MR,MF,NCH)
      CALL RECHAN(R,S,MR,MF,NCH)
D     WRITE(IOUT,412)(R(I),I=1,NCH)
D412  FORMAT(1X,10G12.3)
D     WRITE(IOUT,414)
D414  FORMAT(1H1)
      WRITE(CRT,420)
420   FORMAT(1X,'FILENAME FOR OUTPUT SPECTRUM'/)
      READ(CRT,430)LEN,ASC
430   FORMAT(Q,80A1)
      IF(LEN.LT.1) GOTO 990
      DO 440 I=1,32
      TITLE(I)=DUMT(I)
  440 CONTINUE
      DO 445 I=1,8
      ID(I)=DUMID(I)
  445 CONTINUE
      ELTIME=IDELT
      PLTIME=IDPLT
      ERTIME=IDERT
      PRTIME=IDPRT
      CALL PUT(ASC,R,IVAR,LUF,NCH,A1,B1,C1)
C
990   STOP
      END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME      OFFSET  ATTRIBUTES

S         000006  REAL*4      ARRAY (256)
R         002006  REAL*4      ARRAY (256)
DUMT      004006  INTEGER*2 ARRAY (32)
DUMID     004106  INTEGER*2 ARRAY (8)
AHEAD     004126  REAL*8      ARRAY (10)
ASC       004246  LOGICAL*1 ARRAY (60)
CRT       004342  INTEGER*2 VARIABLE
PERIOD    004352  INTEGER*2 VARIABLE
BLANK     004354  INTEGER*2 VARIABLE
ABLANK    004364  REAL*8      VARIABLE
YES       004374  LOGICAL*1 VARIABLE
LP        004344  INTEGER*2 VARIABLE
IN        004346  INTEGER*2 VARIABLE
IOUT      004350  INTEGER*2 VARIABLE
LUF       004356  INTEGER*2 VARIABLE
BLNK4     004360  REAL*4      VARIABLE
LEN       005424  INTEGER*2 VARIABLE
NCH       005426  INTEGER*2 VARIABLE
GET       000000  REAL*4      PROCEDURE
IVAR      005430  INTEGER*2 VARIABLE
A0        005432  REAL*4      VARIABLE
B0        005436  REAL*4      VARIABLE
C0        005442  REAL*4      VARIABLE
NX        005446  INTEGER*2 VARIABLE
MX        005450  INTEGER*2 VARIABLE
NA        005452  INTEGER*2 VARIABLE
FREEFM    000000  REAL*4      PROCEDURE
I         005454  INTEGER*2 VARIABLE
IDELT     005456  INTEGER*2 VARIABLE
IDPLT     005460  INTEGER*2 VARIABLE
IDERT     005462  INTEGER*2 VARIABLE
IDPRT     005464  INTEGER*2 VARIABLE
KUF       005466  INTEGER*2 VARIABLE
MCH       005470  INTEGER*2 VARIABLE
JVAR      005472  INTEGER*2 VARIABLE
A1        005474  REAL*4      VARIABLE
B1        005500  REAL*4      VARIABLE
C1        005504  REAL*4      VARIABLE
G         005510  REAL*4      VARIABLE
EPS       005514  REAL*4      VARIABLE
MF        005520  INTEGER*2 VARIABLE
MR        005522  INTEGER*2 VARIABLE
REGAIN    000000  REAL*4      PROCEDURE
RECHAN    000000  REAL*4      PROCEDURE
PUT       000000  REAL*4      PROCEDURE

COMMON BLOCK /DATA/     LENGTH 000240

DATA      000000  INTEGER*2 ARRAY (80)
```

```
COMMON BLOCK /FREE/     LENGTH 000340

INTEG     000000  INTEGER*2 ARRAY (16)
REALX     000040  REAL*4      ARRAY (16)
ALPHA     000140  REAL*8      ARRAY (16)
```

```
MIDAS FORTRAN IV   STORAGE MAP


NAME     OFFSET  ATTRIBUTES


COMMON BLOCK /HEAD/     LENGTH 000142

IHDR     000000  INTEGER*2 VARIABLE
TITLE    000002  INTEGER*2 ARRAY (32)
ID       000102  INTEGER*2 ARRAY (8)
ELTIME   000122  REAL*4    VARIABLE
PLTIME   000126  REAL*4    VARIABLE
ERTIME   000132  REAL*4    VARIABLE
PRTIME   000136  REAL*4    VARIABLE
AL*4     PROCEDURE
A        000134  REAL*4    VARIABLE
MIN0     000000  INTEGER*2 PROCEDURE
AI       000140  REAL*4    VARIABLE
AISQ     000144  REAL*4    VARIABLE
B        000150  REAL*4    VARIABLE
AHALF    000154  REAL*4    VARIABLE
AO       000160  REAL*4    VARIABLE
```

AD-A141 539 PROGRAM PREGA PIVOTAL REGRESSION ANALYSIS OF GAMMA-RAY 2/3
SPECTRA FROM NAI(T..(U) NAVAL RESEARCH LAB WASHINGTON
DC G W PHILLIPS ET AL. 06 APR 84 NRL-MR-5275
UNCLASSIFIED SBI-AD-E000 564 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

```
            SUBROUTINE REGAIN(R,S,G,EPS,MR,MF,NCH)
      C     WRITTEN BY G.W. PHILLIPS
      C     NAVAL RESEARCH LABORATORY
      C     LAST REVISED AUGUST 1981 BY G.W.P.
      C
      C     PERFORMS DIGITAL GAIN AND ZERO SHIFT
      C     ON SPECTRAL DATA
      C
            DIMENSION R(1),S(1)
      C
      C     THE EFFECT OF THIS SUBROUTINE IS TO FIRST ADJUST THE
      C     GAIN BY G, AND THEN TO SHIFT THE DATA BY EPS
      C
      C     R=INITIAL DATA ARRAY, S=SHIFTED DATA ARRAY
      C     G=GAIN SHIFT, EPS=ZERO SHIFT, MF=OFFSET FOR CHANNEL 1
      C     MR-NUMBER OF DATA CHANNELS, NCH=MAXIMUM NUMBER OF CHANNELS
      C
            IF(G.NE.1.0.OR.EPS.NE.0.0) GOTO 100
            DO 50 J=1,NCH
       50      S(J)=R(J)
            RETURN
      C
       100  DJ 110 J=1,NCH
       110     S(J)=0.
            MI=MF-1
            AT=G*(MR+MI)+EPS
            MT=INT(AT)
            MT=MIN0(MT,NCH)
            MRI=MR
            MJ=INT(G*MI+EPS)
            MJ=MAX0(MJ,0)
            MF=MJ+1
            MR=MT-MJ
            GINV=1./G
            IF(G.GT.1) GO TO 300
      C
```

```
C
C
C     DISTRIBUTE COUNTS (REBIN)
C
      DO 200 I=1,MRI
         A=G*(I+MI)+EPS
         IF(A.LT.0) GOTO 200
         J=INT(A)
         DJ=A-J
         J=J-MJ
         IF(J.LT.0.OR.J.GT.MR) GOTO 200
         IF(DJ.LT.G) GO TO 160
            IF(J.LT.MR) S(J+1)=S(J+1)+R(I)
            GOTO 200
 160     DR=DJ*GINV*R(I)
         IF(J.LT.MR) S(J+1)=S(J+1)+DR
         IF(J.GT.0) S(J)=S(J)+R(I)-DR
 200  CONTINUE
      JM=J+1
      DJM=DJ
C
C     EXTRAPOLATE ENDS
C
      A0=G*MI+EPS
      IF(A0.LE.0) GOTO 240
      J=INT(A0)
      DJ=A0-J
      J=J-MJ
      IF(J.LT.0) GOTO 240
      S(J+1)=S(J+1)+DJ*GINV*R(1)
      IF(J.LT.1) GOTO 240
      J0=J
      DO 220 J=1,J0
 220     S(J)=GINV*R(1)
C
 240  IF(JM.GT.MR) RETURN
      S(JM)=S(JM)+(1-DJM)*GINV*R(MRI)
      JM=JM+1
      IF(JM.GT.MR) RETURN
      DO 260 J=JM,MR
 260     S(J)=GINV*R(MRI)
      RETURN
```

```
C
C
C     INTERPOLATE BETWEEN MIDPOINTS
C
 300  DO 400 J=1,MR
        A=FLOAT(J+MJ)-0.5
        I=INT((A-EPS)*GINV)
        B=G*(I-0.5)+EPS
        I=I-MI
        IF(A-B.LE.G) GOTO 340
          I=I+1
          B=B+G
 340    IF(I.GT.0) GOTO 350
          S(J)=R(1)*GINV
          GOTO 400
 350    IF(I.LT.MRI) GOTO 360
          S(J)=R(MRI)*GINV
          GOTO 400
 360    DG=(A-B)*GINV
        S(J)=DG*R(I+1)*GINV+(1.-DG)*R(I)*GINV
 400  CONTINUE
      RETURN
      END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

R        000014  REAL*4     PARAMETER ARRAY (1)
S        000016  REAL*4     PARAMETER ARRAY (1)
G        000020  REAL*4     PARAMETER VARIABLE
EPS      000022  REAL*4     PARAMETER VARIABLE
MR       000024  INTEGER*2  PARAMETER VARIABLE
MF       000026  INTEGER*2  PARAMETER VARIABLE
NCH      000030  INTEGER*2  PARAMETER VARIABLE
J        000034  INTEGER*2  VARIABLE
MI       000036  INTEGER*2  VARIABLE
AT       000040  REAL*4     VARIABLE
MT       000044  INTEGER*2  VARIABLE
INT      000000  INTEGER*2  PROCEDURE
MIN0     000000  INTEGER*2  PROCEDURE
MRI      000046  INTEGER*2  VARIABLE
MJ       000050  INTEGER*2  VARIABLE
MAX0     000000  INTEGER*2  PROCEDURE
GINV     000052  REAL*4     VARIABLE
I        000056  INTEGER*2  VARIABLE
A        000060  REAL*4     VARIABLE
DJ       000064  REAL*4     VARIABLE
DR       000070  REAL*4     VARIABLE
JM       000074  INTEGER*2  VARIABLE
DJM      000076  REAL*4     VARIABLE
A0       000102  REAL*4     VARIABLE
J0       000106  INTEGER*2  VARIABLE
FLOAT    000000  REAL*4     PROCEDURE
B        000110  REAL*4     VARIABLE
DG       000114  REAL*4     VARIABLE
```

```
MIDAS FORTRAN IV            11 AUG 1983  5:25:16 PM       PAGE 001

            SUBROUTINE RECHAN(R,S,MR,MF,NCH)
      C     WRITTEN BY G.W.PHILLIPS, AUGUST 1981
      C
      C     SHIFTS DATA IN ARRAY S BY MF-1 CHANNELS
      C     AND STORES THE SHIFTED DATA IN ARRAY R
            DIMENSION R(256),S(256)
      C
      100   K=MF-1
            MF=1
            MR=MR+K
            IF(MP LT.0) MR=0
            IF(MR.GT.NCH) MR=NCH
            J=1
            IF(K.LT.1) GOTO 120
            J=K+1
            DO 110 I=1,K
      110   R(I)=0.
      120   DO 130 I=J,MR
      130   R(I)=S(I-K)
            L=MR+1
            IF(L.GT.NCH) RETURN
            DO 140 I=L,NCH
      140   R(I)=0.
            RETURN
            END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

R        000014  REAL*4     PARAMETER ARRAY (256)
S        000016  REAL*4     PARAMETER ARRAY (256)
MR       000020  INTEGER*2 PARAMETER VARIABLE
MF       000022  INTEGER*2 PARAMETER VARIABLE
NCH      000024  INTEGER*2 PARAMETER VARIABLE
K        000026  INTEGER*2 VARIABLE
J        000030  INTEGER*2 VARIABLE
I        000032  INTEGER*2 VARIABLE
L        000034  INTEGER*2 VARIABLE
```

```
      C
      C
      C
      C        COPYRIGHT (C) 1977 BY NUCLEAR DATA INCORPORATED
      C        SCHAUMBURG, ILLINOIS 60196
      C
      C
      C        THIS DOCUMENTATION IS FURNISHED INCLUDING THE COPYRIGHT
      C        NUCLEAR DATA INCORPORATED. IT MUST NOT BE REMOVED AND MAY NOT
      C        BE REPRODUCED OR REDISTRIBUTED WITHOUT THE WRITTEN EXPLICIT
      C        PERMISSION OF NUCLEAR DATA.
      C
      C        WHEREBY NUCLEAR DATA CORPORATION ASSUMES THE SUPPORT
      C        RESPONSIBILITIES WHEN PROPERLY APPLIED TO ND6600 (MIDAS)
      C        SYSTEMS.
      C
      C         NUCLEAR DATA CORPORATION ASSUMES NO RESPONSIBILITY FOR USE
      C        OF AFORMENTIONED SOFTWARE IN EQUIPMENT EXCLUDED FROM THE
      C        ND6600-SERIES EQUIPMENT SUPPLIED BY NUCLEAR DATA INCORPORATED.
      C
      C      SEDIT   MOD 44-0436-01     10 MAR 77     L.HOLMES
      C      THIS IS THE PREGA LIBRARY FILE EDITOR
      C      IT BUILDS A LIST OF THE STANDARD ISOTOPE SPECTRA
      C      TO BE USED FROM THE DISK
      C
      C
      C      FILE STRUCTURE
      C
      C   RECORD :  PARAMETER
      C
      C      1   NUMBER OF STANDARDS IN LIBRARY
      C      2   NUMBER OF CHANNELS PER STANDARD
      C      3   NUMBER OF ITERATIONS FOR GAIN THRESHOLD SHIFT
      C      4   BACKGROUND SUBTRACT ?
      C      5   INITIAL CHANNEL FOR COMPUTATION
      C      6   FINAL CHANNEL FOR COMPUTATION
      C   7-39   SPACE RESERVED FOR FILE EXPANSION
      C     40   COUNTING TIME FOR BACKGROUND
      C     41   REJECTION COEFFICIENT.
      C  42-49   ID FOR LIBRARY BACKGROUND
      C  50-64   SPACE RESERVED FOR FILE EXPANSION
      C
      C
      C  65-96,97-128,....  LIST OF STANDARDS WITH ASSOCIATED PARAMETERS
      C          I.E.
      C
      C   RECORD :  PARAMETER
      C
      C     65   EIGHT CHARACTER NAME OF STANDARD
      C  66-73   FILE.ELEMENT ID
      C     74   HALF-LIFE
      C     75   COUNTING TIME OF SAMPLE


      C     76   ACTIVITY OF SAMPLE
      C
      C
             INTEGER ANS,YES
             DATA YES,NO/1HY,1HN/
```

```
            INTEGER INT(26),MLI(40)
            LOGICAL*1 A(32)
            DATA LN/1HN/
            DATA LI,LD,LA,LP/1HI,1HD,1HA,1HL/
            INTEGER BB,CC
            DATA BB,CC/1HB,1HC/
            INTEGER BS,BT,RC,BA
            DATA NC,IT,BS,IS,IE,LL,LS,BT,RC,BA
          1/2HNC,2HIT,2HBS,2HIS,2HIE,2HLL,2HLS,2HBT,2HRC,2HBA/
            INTEGER Y,D,H
            DATA Y,D,H,MM/1HY,1HD,1HH,1HM/
            INTEGER EDT
            EDT=0
            L=5
            N=0
            LUN=9
            LUI=11
            DO 1200 I=1,40
      1200  MLI(I)=I
            DEFINE FILE LUN(4096,2,U,IVAR)
            DEFINE FILE LUI(4096,2,U,IVAR)
            IVAR=1
            READ(LUN'1,END=500,ERR=600)M
            WRITE(5,1)
    1       FORMAT(1X,'PREGA LIBRARY FILE EDITOR',/)
    10000   CONTINUE
    1999    WRITE(5,2000)
    2000    FORMAT(1X,'HEADER: BUILD(B), CHANGE(C), LIST(L) ',$)
            READ(5,5000)LENGTH,ANS
            IF(LENGTH.LE.0)GO TO 15
            IF(ANS.EQ.BB)GO TO 2001
            IF(ANS.EQ.CC)GO TO 2500
            IF(ANS.EQ.LP)GO TO 6500
            GO TO 10000
    2001    ASSIGN 2050 TO K
    2040    WRITE(5,2041)
    2041    FORMAT(' ','(NC) NUMBER OF CHANNELS PER STANDARD (MAX-512) ',$)
    2042    READ(5,5001)LENGTH,FNUM
            IF(LENGTH.LE.0)GO TO 10000
            IF(FNUM.GT.512.)FNUM=512.
            IF(FNUM.LE.1.0)FNUM=1.0
            NUM=FNUM
            WRITE(LUN'2)NUM
            GO TO K
    2050    ASSIGN 2060 TO K
            WRITE(5,2051)
    2051    FORMAT(' ','(IT) NUMBER OF ITERATIONS FOR THRESHOLD & GAIN CALC.
          1 ',$)
    2052    READ(5,5001)LENGTH,FNUM
            IF(LENGTH.LE.0)GO TO 10000
            IF(FNUM.LE.1.0)FNUM=1.0
            NUM=FNUM
            WRITE(LUN'3)NUM
            GO TO K
    2060    ASSIGN 2070 TO K
            WRITE(5,2061)
    2061    FORMAT(' ','(BS) BACKGROUND SUBTRACT ? ',$)
```

```
0067  2062  READ(5,5000)LENGTH,ANS
0068        IF(LENGTH.LE.0)GO TO 10000
0070        NUM=0
0071        IF(ANS.EQ.YES)NUM=1
0073        WRITE(LUN'4)NUM
0074        GO TO K
0075  2070  ASSIGN 2080 TO K
0076        WRITE(5,2071)
0077  2071  FORMAT(' ','(IS) INITIAL CHANNEL FOR COMPUTATION ',$)
0078  2072  READ(5,5001)LENGTH,FNUM
0079        IF(LENGTH.LE.0)GO TO 10000
0081        IF(FNUM.GT.512.)FNUM=512.
0083        IF(FNUM.LE.1.0)FNUM=1.0
0085        NUM=FNUM
0086        WRITE(LUN'5)NUM
0087        GO TO K
0088  2080  ASSIGN 2110 TO K
0089        WRITE(5,2081)
0090  2081  FORMAT(' ','(IE) FINAL CHANNEL FOR COMPUTATION ',$)
0091  2082  READ(5,5001)LENGTH,FNUM
0092        IF(LENGTH.LE.0)GO TO 10000
0094        IF(FNUM.GT.512.)FNUM=512.
0096        IF(FNUM.LE.1.0)FNUM=1.0
0098        NUM=FNUM
0099        WRITE(LUN'6)NUM
0100        GO TO K
0101  2110  ASSIGN 2120 TO K
0102        WRITE(5,2111)
0103  2111  FORMAT(' ','(BT) COUNTING TIME OF BACKGROUND (SECS.) ',$)
0104  2112  READ(5,5001)LENGTH,FNUM
0105        IF(LENGTH.LE.0)GO TO 10000
0107        IF(FNUM.LE.1.0)FNUM=1.0
0109        WRITE(LUN'40)FNUM
0110        GO TO K
0111  2120  ASSIGN 2130 TO K
0112        WRITE(5,2121)
0113  2121  FORMAT(' ','(RC) REJECTION COEFICIENT ',$)
0114  2122  READ(5,5001)LENGTH,FNUM
0115        IF(LENGTH.LE.0)GO TO 10000
0117        WRITE(LUN'41)FNUM
0118        GO TO K
0119  2130  ASSIGN 2140 TO K
0120        WRITE(5,2131)
0121  2131  FORMAT(' ','(BA) BACKGROUND SPECTRUM (FILE.ELEMENT(,DEVICE)) ',/)
0122  2132  READ(5,4)LENGTH,(A(I),I=1,32)
0123        IF(LENGTH.LE.0)GO TO 10000
0125        IVAR=42
0126        DO 2133 I=1,32,4
0127  2133  WRITE(LUN'IVAR)A(I),A(I+1),A(I+2),A(I+3)
0128        GO TO K
0129  2140  GO TO 1999
0130  2500  ASSIGN 2500 TO K
0131        WRITE(5,2501)
0132  2501  FORMAT(' ','WHICH PARAMETER DO YOU WISH TO MODIFY ? ',$)
0133        READ(5,5007)LENGTH,ANS
0134        IF(LENGTH.LE.0)GO TO 10000
0136        IF(ANS.EQ.NC)GO TO 6000
```

```
      IF(ANS.EQ.IT)GO TO 6001
      IF(ANS.EQ.BS)GO TO 6002
      IF(ANS.EQ.IS)GO TO 6003
      IF(ANS.EQ.IE)GO TO 6004
      IF(ANS.EQ.BT)GO TO 6007
      IF(ANS.EQ.RC)GO TO 6008
      IF(ANS.EQ.BA)GO TO 6009
      GO TO 2500
6000  READ(LUN'2)NUM
      WRITE(L,2041)
      WRITE(L,5004)NUM
      IF(EDT.EQ.1)GO TO 7000
      GO TO 2042
6001  READ(LUN'3)NUM
      WRITE(L,2051)
      WRITE(L,5004)NUM
      IF(EDT.EQ.1)GO TO 7000
      GO TO 2052
6002  READ(LUN'4)NUM
      WRITE(L,2061)
      ANS=NO
      IF(NUM.EQ.1)ANS=YES
      WRITE(L,5008)ANS
      IF(EDT.EQ.1)GO TO 7000
      GO TO 2062
6003  READ(LUN'5)NUM
      WRITE(L,2071)
      WRITE(L,5004)NUM
      IF(EDT.EQ.1)GO TO 7000
      GO TO 2072
6004  READ(LUN'6)NUM
      WRITE(L,2081)
      WRITE(L,5004)NUM
      IF(EDT.EQ.1)GO TO 7000
      GO TO 2082
6007  READ(LUN'40)FNUM
      WRITE(L,2111)
      WRITE(L,5005)FNUM
      IF(EDT.EQ.1)GO TO 7000
      GO TO 2112
6008  READ(LUN'41)FNUM
      WRITE(L,2121)
      WRITE(L,5005)FNUM
      IF(EDT.EQ.1)GO TO 7000
      GO TO 2122
6009  IVAR=42
      DO 2600 I=1,32,4
2600  READ(LUN'IVAR)A(I),A(I+1),A(I+2),A(I+3)
      WRITE(L,2131)
      WRITE(L,5006)(A(I),I=1,32)
      IF(EDT.EQ.1)GO TO 7000
      GO TO 2132
C  HEADER LISTER
6500  CONTINUE
      L=6
      EDT=1
      WRITE(L,6501)
```

```
0210  6501  FORMAT('1','            PREGA LIBRARY HEADER',//)
0211        I=0
0212  7000  I=I+1
0213        IF(I.GT. 8)GO TO 6010
0215        GO TO (6000,6001,6002,6003,6004,          6007,6008,6009) ,I
0216  6010  L=5
0217        EDT=0
0218        WRITE(6,411)
0219        WRITE(6,411)
0220        GO TO 1999
0221  3000  CONTINUE
0222  15    CONTINUE
0223        WRITE(5,16)
0224  16    FORMAT(1X,'LIST OF STANDARDS')
0225        WRITE(5,111)
0226  111   FORMAT(1X,'NEW FILE(N),INSERT(I),DELETE(D),APPEND(A),LIST(L) ',$)
0227        IVAR=1
0228        N=0
0229        EDT=0
0230        READ(5,2)LENGTH,ANS
0231  2     FORMAT(Q,A1)
0232        IF(LENGTH.EQ.0)GO TO 1000
0234        IF(ANS.EQ.LN)GO TO 25
0236        IF(ANS.EQ.LI)GO TO 100
0238        IF(ANS.EQ.LD)GO TO 200
0240        IF(ANS.EQ.LA)GO TO 300
0242        IF(ANS.EQ.LP)GO TO 400
0244        GO TO 15
0245  25    CONTINUE
0246        IVAR=65+N*16
0247        WRITE(5,20)
0248  20    FORMAT(1X,'ISOTOPE NAME (8 CHARACHTERS MAX) ',$)
0249        READ(5,21)LENGTH,(A(I),I=1,8)
0250  21    FORMAT(Q,8A1)
0251        IF(LENGTH.EQ.0)GO TO 15
0253        DO 22 I=1,8,4
0254  22    WRITE(LUN'IVAR)A(I),A(I+1),A(I+2),A(I+3)
0255        WRITE(5,3)
0256  3     FORMAT(1X,'FILE.ELEMENT(,DEVICE) ',$)
0257        READ(5,4)LENGTH,(A(I),I=1,32)
0258  4     FORMAT(Q,32A1)
0259        IF(LENGTH.EQ.0)GO TO 15
0261        DO 10 I=1,32,4
0262  10    WRITE(LUN'IVAR,END=500,ERR=600)A(I),A(I+1),A(I+2),A(I+3)
0263        WRITE(5,5)
0264  5     FORMAT(1X,'HALF-LIFE,UNITS(Y,D,H,M,S) ',$)
0265        READ(5,6)LENGTH,HL,K
0266  6     FORMAT(Q,E12.0,1A1)
0267        IF(LENGTH.EQ.0)GO TO 15
0269        IF(K.EQ.Y)HL=HL*31536000.
0271        IF(K.EQ.D)HL=HL*86400.
0273        IF(K.EQ.H)HL=HL*3600.
0275        IF(K.EQ.MM)HL=HL*60.
0277        WRITE(LUN'IVAR)HL
0278        WRITE(5,7)
0279  7     FORMAT(1X,'COUNTING TIME OF STANDARD (SECS.) ',$)
0280        READ(5,5001)LENGTH,SEC
```

```
          IF(LENGTH.EQ.0)GO TO 15
          IF(SEC.LE.1.0)SEC=1.0
          WRITE(LUN'IVAR)SEC
          WRITE(5,9)
    9     FORMAT(1X,'ACTIVITY (MAY BE SET TO 1) ',$)
          READ(5,5001)LENGTH,ACT
          IF(LENGTH.EQ.0)GO TO 15
          WRITE(LUN'IVAR)ACT
          IF(EDT.EQ.1)GO TO 15
          N=N+1
          M=N
          WRITE(LUN'1)M
          GO TO 25
    100   WRITE(5,101)
    101   FORMAT(1X,'NUMBER OF ENTRY TO BE INSERTED ',$)
          READ(5,102)FNUM
    102   FORMAT(F10.0)
          NUM=FNUM
          N=M
          IF(N.LT.NUM)GO TO 15
          M=M+1
          N=M
          EDT=1
    103   IF(N.LT.NUM)GO TO 25
          IVAR=65+(N-1)*16
          DO 104 I=1,26,2
    104   READ(LUN'IVAR)INT(I),INT(I+1)
          IVAR=IVAR+3
          DO 105 I=1,26,2
    105   WRITE(LUN'IVAR)INT(I),INT(I+1)
          N=N-1
          GO TO 103
    C
    200   WRITE(5,201)
    201   FORMAT(1X,'NUMBER OF ENTRY TO BE DELETED ',$)
          READ(5,202)FNUM
    202   FORMAT(F10.0)
          NUM=FNUM
          IF(NUM.GT.M)GO TO 15
          IF(NUM.EQ.M)GO TO 206
          IF(M.LE.0)GO TO 206
    203   CONTINUE
          IVAR=NUM*16+65
          DO 204 I=1,26,2
    204   READ(LUN'IVAR,END=500,ERR=600)INT(I),INT(I+1)
          IVAR=IVAR-29
          DO 205 I=1,26,2
    205   WRITE(LUN'IVAR,END=500,ERR=600) INT(I),INT(I+1)
          NUM=NUM+1
          IF(NUM.LT.M)GO TO 203
    206   CONTINUE
          M=M-1
          GO TO 15
    300   N=M
          GO TO 25
    400   IF(M.EQ.0)GO TO 15
    C*********************************
```

```
0347    1008 WRITE(5,1010)
0348    1010 FORMAT(' ','DO YOU WISH TO PRINT FULL STANDARD LIST?',$)
0349         READ(5,2)LENGTH,ANS
0350         IF(LENGTH.LE.0) GO TO 1008
0352         IF(ANS.EQ.YES)GO TO 1020
0354         READ(LUI'29) MLA
       D     WRITE(6,7100) MLA
       D7100 FORMAT(' ',' MLA=',I6)
       D     WRITE(5,7105)
       D7105 FORMAT(' ','  IVAR=?',$)
       D     READ(5,7106) IVAR
       D7106 FORMAT(I2)
0355         IVAR=44
0356         DO 1014 I=1,MLA
0357         READ(LUI'IVAR) MLI(I)
       D     WRITE(6,7110) IVAR,I,MLI(I)
       D7110 FORMAT(' ',' IVAR=',I3,'  MLI(',I2,')=',I6)
0358    1014 CONTINUE
0359         WRITE(6,1016)
0360    1016 FORMAT('1','           PREGA LIBRARY STANDARDS USED',//)
0361         GO TO 1024
0362    1020 MLA=M
       D     WRITE(6,7120) MLA
       D7120 FORMAT(' ','  MLA=M=',I4)
0363         WRITE(6,401)
0364   401   FORMAT('1','            PREGA LIBRARY STANDARDS',//)
0365    1024 DO 1090 JI=1,MLA
0366         IVAR=65+16*(MLI(JI)-1)
0367         DO 415 I=1,8,4
0368   415   READ(LUN'IVAR)A(I),A(I+1),A(I+2),A(I+3)
0369         WRITE(6,425)MLI(JI),(A(I),I=1,8)
0370   425   FORMAT(1X,'STANDARD # ',I2,' : ',8A1)
0371         DO 404 I=1,32,4
0372   404   READ(LUN'IVAR)A(I),A(I+1),A(I+2),A(I+3)
0373         WRITE(6,3)
0374         WRITE(6,405)(A(I),I=1,32)
0375   405   FORMAT('+', ': ',32A1)
0376         READ(LUN'IVAR)SEC
0377         WRITE(6,406)SEC
0378   406   FORMAT(1X,'HALF-LIFE (SECS):',1PE12.4)
0379         READ(LUN'IVAR)HL
0380         WRITE(6,407)HL
0381   407   FORMAT(1X,'COUNTING TIME OF STANDARD (SECS):',1PE12.4)
0382         READ(LUN'IVAR)ACT
0383         WRITE(6,408)ACT
0384   408   FORMAT(1X,'ACTIVITY:',1PE12.4,//)
0385    1090 CONTINUE
       C     ****************************************
0386   410   WRITE(6,411)
0387         WRITE(6,411)
0388         WRITE(6,411)
0389         WRITE(6,411)
0390         WRITE(6,411)
0391         WRITE(6,411)
0392         WRITE(6,411)
0393         WRITE(6,411)
0394   411   FORMAT('+','
      GO TO 15
500   WRITE(5,501)
501   FORMAT(1X,'FILE NOT LARGE ENOUGH')
      GO TO 1000
600   WRITE(5,601)
601   FORMAT(1X,'FILE SET UP NOT CORRECT')
      GO TO 1000
5000  FORMAT(Q,A1)
5001  FORMAT(Q,E12.0)
5003  FORMAT(Q,32A1)
5004  FORMAT('+',': ',I4,/)
5005  FORMAT('+',': ',G12.6,/)
5006  FORMAT(1X,'      : ',32A1,/)
5007  FORMAT(Q,A2)
5008  FORMAT('+',': ',1A1,/)
1000  IF(M.GE.40)M=40
      WRITE(LUN'1)M
      END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

INT      000006  INTEGER*2 ARRAY (26)
MLI      000072  INTEGER*2 ARRAY (40)
A        000212  LOGICAL*1 ARRAY (32)
ANS      003000  INTEGER*2 VARIABLE
YES      000252  INTEGER*2 VARIABLE
BB       000270  INTEGER*2 VARIABLE
CC       000272  INTEGER*2 VARIABLE
BS       000300  INTEGER*2 VARIABLE
BT       000312  INTEGER*2 VARIABLE
RC       000314  INTEGER*2 VARIABLE
BA       000316  INTEGER*2 VARIABLE
Y        000320  INTEGER*2 VARIABLE
D        000322  INTEGER*2 VARIABLE
H        000324  INTEGER*2 VARIABLE
EDT      003002  INTEGER*2 VARIABLE
NO       000254  INTEGER*2 VARIABLE
LN       000256  INTEGER*2 VARIABLE
LI       000260  INTEGER*2 VARIABLE
LD       000262  INTEGER*2 VARIABLE
LA       000264  INTEGER*2 VARIABLE
LP       000266  INTEGER*2 VARIABLE
NC       000274  INTEGER*2 VARIABLE
IT       000276  INTEGER*2 VARIABLE
IS       000302  INTEGER*2 VARIABLE
IE       000304  INTEGER*2 VARIABLE
LL       000306  INTEGER*2 VARIABLE
LS       000310  INTEGER*2 VARIABLE
MM       000326  INTEGER*2 VARIABLE
L        003004  INTEGER*2 VARIABLE
N        003006  INTEGER*2 VARIABLE
LUN      003010  INTEGER*2 VARIABLE
LUI      003012  INTEGER*2 VARIABLE
I        003014  INTEGER*2 VARIABLE
IVAR     003016  INTEGER*2 VARIABLE
M        003020  INTEGER*2 VARIABLE
LENGTH   003022  INTEGER*2 VARIABLE
K        003024  INTEGER*2 VARIABLE
FNUM     003026  REAL*4    VARIABLE
NUM      003032  INTEGER*2 VARIABLE
HL       003034  REAL*4    VARIABLE
SEC      003040  REAL*4    VARIABLE
ACT      003044  REAL*4    VARIABLE
MLA      003050  INTEGER*2 VARIABLE
JI       003052  INTEGER*2 VARIABLE
```

```
C     PROGRAM UEDIT
C     MODIFIED 8 JUNE 1981 BY G.PHILLIPS
C
C       COPYRIGHT (C) 1977 BY NUCLEAR DATA INCORPORATED
C       SCHAUMBURG, ILLINOIS 60196
C
C       THIS DOCUMENTATION IS FURNISHED INCLUDING THE COPYRIGHT
C       NUCLEAR DATA INCORPORATED. IT MUST NOT BE REMOVED AND MAY NOT
C       BE REPRODUCED OR REDISTRIBUTED WITHOUT THE WRITTEN EXPLICIT
C       PERMISSION OF NUCLEAR DATA.
C
C       WHEREBY NUCLEAR DATA CORPORATION ASSUMES THE SUPPORT
C       RESPONSIBILITIES WHEN PROPERLY APPLIED TO ND6600 (MIDAS)
C       SYSTEMS.
C
C        NUCLEAR DATA CORPORATION ASSUMES NO RESPONSIBILITY FOR USE
C       OF AFORMENTIONED SOFTWARE IN EQUIPMENT EXCLUDED FROM THE
C       ND6600-SERIES EQUIPMENT SUPPLIED BY NUCLEAR DATA INCORPORATED.
C
C     UEDIT    MOD 44-0435-01   10 MAR 77   L.HOLMES
C     THIS IS THE NAI UNKNOWN LIBRARY EDITOR.
C     IT SETS UP THE ANALYSIS PARAMETERS FOR
C     THE UNKNOWN SAMPLE SPECTRUM.
C
C
C     FILE STRUCTURE
C
C  RECORD :   PARAMETER
C
C   1-8   ID OF SAMPLE
C    10   GAIN SHIFT RATIO
C    11   THRESHOLD SHIFT
C    12   BACKGROUND SUPPLIED ?
C    13   BACKGROUND SUBTRACT ?
C    14   COUNTING TIME FOR BACKGROUND
C    15   COUNTING TIME FOR UNKNOWN SAMPLE
C    16   VOLUME REDUCTION FACTOR
C    17   DECAY TIME
C    18   RESULT MULTIPLICATION FACTOR
C 19-27   ID OF SAMPLE BACKGROUND
C    28   WEIGHTING FACTOR DETERMINATION
C    29   NUMBER OF STANDARDS USED
C    30   F-TEST REJECTION COEFFICIENT
C    31   WEIGHTING FACTOR CALCULATION
C    32   LIBRARY REJECTION COEFFICIENT APPLIED ?
C          THIS REJECTION COEFFICIENT IS IN SEDIT.
C    33   BACKGROUND REGRESSION?
C    34   PIVOT ON BACKGROUND?
C    35   OUTPUT RESIDUALS?
C 36-42    ID OF OUTPUT FILE
C 43,44,...    ORDER OF OUTPUT FILE
```

```
C
C
      INTEGER  CC
      INTEGER EDT
      INTEGER UN,OP,CB,BG,BS,BT,UT,VR,DT,MF,SB,WD,NS,F0,WF,RC,BR,BP,ANS,
     1YES,CRT,RO,RS
      DATA UN,OP,CB,BG,BS,BT,VR,DT,MF,SB,WD,NS,F0,WF,RC,BR,BP,RO,RS
     1/2HUN,2HOP,2HCB,2HBG,2HBS,2HBT,2HVR,2HDT,2HMF,2HSB,2HWD,2HNS,2HF0,
     22HWF,2HRC,2HBR,2HBP,2HRO,2HRS/
      INTEGER GN,TH
      DATA GN,TH/2HGN,2HTH/
      INTEGER OR
      DATA OR/2HOR/
      DATA UT/2HUT/
      DATA NN,CC,LL,YES,NO/1HN,1HC,1HL,1HY,1HN/
      INTEGER CT
      DATA CT/2HCT/
      INTEGER BA
      DATA BA/2HBA/
      LOGICAL*1 A(32)
      LUN=11
      DEFINE FILE LUN(4096,2,U,IVAR)
      CRT=5
      READ(LUN'1)M
      WRITE(CRT,1)
1     FORMAT(1X,'PREGA UNKNOWN FILE EDITOR',//)
10000 CONTINUE
1500  WRITE(5,4)
4     FORMAT(1X,'NEW FILE(N), CHANGE(C), LIST(L) ',$)
      READ(CRT,1002)LENGTH,ANS
      IF(LENGTH.LE.0)GO TO 11000
      IVAR=2
      IF(ANS.EQ.CC)GO TO 450
      IF(ANS.EQ.LL)GO TO 900
      IF(ANS.EQ.NN)GO TO 49
      GO TO 1500
      C
      C BUILD INDIVIDUAL FILES
0036  49    CONTINUE
0037        ASSIGN 60 TO K
0038  50    WRITE(CRT,51)
0039  51    FORMAT(1X,'(UN) UNKNOWN SPECTRUM: FILE.ELEMENT(,DEVICE) ',/)
0040  52    READ(CRT,1001)LENGTH,(A(I),I=1,32)
0041        IF(LENGTH.LE.0)GO TO 10000
0043        DO 55 I=1,32,4
0044  55    WRITE(LUN'IVAR)A(I),A(I+1),A(I+2),A(I+3)
0045        GO TO K
0046  60    ASSIGN 70 TO K
0047        WRITE(CRT,61)
0048  61    FORMAT(1X,'(GN) GAIN SHIFT RATIO ',$)
0049  62    READ(CRT,1003)LENGTH,FNUM
0050        IF(LENGTH.LE.0)GO TO 10000
0052        IF(FNUM.LE.(1.0E-5))FNUM=1.0
0054        WRITE(LUN'IVAR)FNUM
0055        GO TO K
0056  70    ASSIGN 80 TO K
0057        WRITE(CRT,71)
0058  71    FORMAT(1X,'(TH) THRESHOLD SHIFT ',$)
0059  72    READ(CRT,1003)LENGTH,FNUM
0060        IF(LENGTH.LE.0)GO TO 10000
0062        WRITE(LUN'IVAR)FNUM
0063        GO TO K
0064  80    ASSIGN 90 TO K
0065        WRITE(CRT,81)
0066  81    FORMAT(1X,'(BG) BACKGROUND SUPPLIED ? ',$)
0067  82    READ(CRT,1002)LENGTH,ANS
0068        IF(LENGTH.LE.0)GO TO 10000
0070        NUM=0
0071        IF(ANS.EQ.YES)NUM=1
0073        WRITE(LUN'IVAR)NUM
0074        GO TO K
0075  90    ASSIGN 100 TO K
0076        WRITE(CRT,91)
0077  91    FORMAT(1X,'(BS) SUBTRACT BACKGROUND ? ',$)
0078  92    READ(CRT,1002)LENGTH,ANS
0079        IF(LENGTH.LE.0)GO TO 10000
0081        NUM=0
0082        IF(ANS.EQ.YES)NUM=1
0084        WRITE(LUN'IVAR)NUM
0085        GO TO K
```

```
      C
      100   ASSIGN 110 TO K
            WRITE(CRT,101)
      101   FORMAT(1X,'(BT) COUNTING TIME FOR BACKGROUND (SECS) ',$)
      102   READ(CRT,1003) LENGTH,FNUM
            IF(LENGTH.LE.0)GO TO 10000
            IF(FNUM.LE.1.0)FNUM=1.0
            WRITE(LUN'IVAR)FNUM
            GO TO K
      110   ASSIGN 120 TO K
            WRITE(CRT,111)
      111   FORMAT(1X,'(CT) COUNTING TIME FOR UNKNOWN SAMPLE ',$)
      112   READ(CRT,1003)LENGTH,FNUM
            IF(LENGTH.LE.0)GO TO 10000
            IF(FNUM.LE.1.0)FNUM=1.0
            WRITE(LUN'IVAR)FNUM
            GO TO K
      120   ASSIGN 130 TO K
            WRITE(CRT,121)
      121   FORMAT(1X,'(VR) VOLUME REDUCTION (OR INVERSE OF DILUTION)',/,1X,
           1'      FACTOR ',$)
      122   READ(CRT,1003)LENGTH,FNUM
            IF(LENGTH.LE.0)GO TO 10000
            IF(FNUM.LE.(1.0E-5))FNUM=1.0
            WRITE(LUN'IVAR)FNUM
            GO TO K
      130   ASSIGN 140 TO K
            WRITE(CRT,131)
      131   FORMAT(1X,'(DT) DECAY TIME ',$)
      132   READ(CRT,1003)LENGTH,FNUM
            IF(LENGTH.LE.0)GO TO 10000
            WRITE(LUN'IVAR)FNUM
            GO TO K
      140   ASSIGN 150 TO K
            WRITE(CRT,141)
      141   FORMAT(1X,'(MF) RESULT MULTIPLICATION FACTOR ',$)
      142   READ(CRT,1003)LENGTH,FNUM
            IF(LENGTH.LE.0)GO TO 10000
            WRITE(LUN'IVAR)FNUM
            GO TO K
      150   ASSIGN 180 TO K
            WRITE(CRT,151)
      151   FORMAT(1X,'(BA) BACKGROUND SPECTRUM: FILE.ELEMENT(,DEVICE) ',/)
      152   READ(CRT,1001)LENGTH,(A(I),I=1,32)
            IF(LENGTH.LE.0)GO TO 10000
            DO 153 I=1,32,4
      153   WRITE(LUN'IVAR)A(I),A(I+1),A(I+2),A(I+3)
            GO TO K
```

```
C
160   ASSIGN 170 TO K
162   WRITE(CRT,161)
161   FORMAT(1X,'ANALYSIS OPTIONS ')
      GO TO K
170   ASSIGN 180 TO K
      WRITE(CRT,171)
171   FORMAT(1X,'(SB) SUBTRACT BACKGROUND ? ',$)
172   READ(CRT,1002)LENGTH,ANS
      IF(LENGTH.LE.0)GO TO 10000
      WRITE(LUN'IVAR)ANS
      GO TO K
180   ASSIGN 190 TO K
      IVAR=IVAR+1
      WRITE(CRT,181)
181   FORMAT(1X,'(WD) HOW ARE WEIGHTING FACTORS TO BE DETERMINED ?',/,1X
     1,
     2'      0: BASED ON ACTUAL COUNTS/CHANNEL.',/,1X,
     3'      1: BASED ON CALCULATED COUNTS/CHN ',$)
182   READ(CRT,1003)LENGTH,FNUM
      IF(LENGTH.LE.0)GO TO 10000
      NUM=0
      IF(FNUM.EQ.1.0)NUM=1
      WRITE(LUN'IVAR)NUM
      GO TO K
190   ASSIGN 200 TO K
      WRITE(CRT,191)
191   FORMAT(1X,'(NS) NUMBER OF ISOTOPES USED FROM THE STANDARD LIBRARY
     1?'$)
192   READ(CRT,1003)LENGTH,FNUM
      IF(LENGTH.LE.0)GO TO 10000
      NUM=FNUM
      IF(NUM.GE.15)NUM=15
      IF(NUM.LE.1)NUM=1
      WRITE(LUN'IVAR)NUM
      GO TO K
```

```
C
200   ASSIGN 210 TO K
      WRITE(CRT,201)
201   FORMAT(1X,'(F0) F-TEST REJECTION COEFFICIENT.',$)
202   READ(CRT,1003)LENGTH,FNUM
      IF(LENGTH.LE.0)GO TO 10000
      WRITE(LUN'IVAR)FNUM
      GO TO K
210   ASSIGN 220 TO K
      WRITE(CRT,211)
211   FORMAT(1X,'(WF) WEIGHTING FACTOR CALCULATION.'
     1,/,1X,'     -1: (COUNTS)**-1 '
     2,/,1X,'      0: (SIGMA)**-2 '
     3,/,1X,'      1: 1 ',$)
212   READ(CRT,1003)LENGTH,FNUM
      IF(LENGTH.LE.0)GO TO 10000
      NUM=1
      IF(FNUM.EQ.0.0)NUM=0
      IF(FNUM.EQ.(-1.0))NUM=-1
      NUM=FNUM
      WRITE(LUN'IVAR)NUM
      GO TO K
220   ASSIGN 230 TO K
      WRITE(CRT,221)
221   FORMAT(1X,'(RC) REJECTION COEFICIENT APPLIED ? ',$)
222   READ(CRT,1002)LENGTH,ANS
      IF(LENGTH.LE.0)GO TO 10000
      NUM=0
      IF(ANS.EQ.YES)NUM=1
      WRITE(LUN'IVAR)NUM
      GO TO K
230   ASSIGN 240 TO K
      WRITE(CRT,231)
231   FORMAT(1X,'(BR) BACKGROUND REGRESSION ?'
     1,/,6X,' 0: TRIAL SET = CHOSEN STANDARDS'
     2,/,6X,' 1: TRIAL SET = BACKGROUND ONLY',$)
232   READ(CRT,1003)LENGTH,FNUM
      IF(LENGTH.LE.0)GO TO 10000
      NUM=FNUM
      WRITE(LUN'IVAR)NUM
      GO TO K
      C
      240   ASSIGN 243 TO K
            WRITE(CRT,241)
      241   FORMAT(1X,'(BP) PIVOT ON BACKGROUND SPECTRUM ALLOWED?'
           1,/,6X,' 0: NO PIVOT ON BACKGROUND ALLOWED'
           2,/,6X,' 1: ALLOW PIVOT ON BACKGROUND SPECTRUM',$)
      242   READ(CRT,1003)LENGTH,FNUM
            IF(LENGTH.LE.0)GO TO 10000
            NUM=FNUM
            WRITE(LUN'IVAR)NUM
            GO TO K
      243   ASSIGN 246 TO K
            WRITE(CRT,244)
      244   FORMAT(1X,'(RO) OUTPUT RESIDUALS ?',$)
      245   READ(CRT,1002)LENGTH,ANS
            IF(LENGTH.LE.0) GOTO 10000
            NUM=0
            IF(ANS.EQ.YES) NUM=1
            WRITE(LUN'IVAR)NUM
            GOTO K
      246   ASSIGN 250 TO K
            WRITE(CRT,247)
      247   FORMAT(1X,'(RS) RESIDUAL SPECTRUM: FILE.ELEMENT(,DEVICE)')
      248   READ(CRT,1001)LENGTH,(A(I),I=1,32)
            IF(LENGTH.LE.0) GOTO 10000
            DO 249 I=1,32,4
      249   WRITE(LUN'IVAR)A(I),A(I+1),A(I+2),A(I+3)
            GOTO K
      250   ASSIGN 10000 TO K
            IVAR=IVAR-6
            READ(LUN'IVAR)NUM
            ITT=NUM
            IVAR=IVAR+5
            WRITE(CRT,251)
      251   FORMAT(1X,'(OR) NUMBER OF LIBRARY STANDARD IN ORDER OF DESIRED ',
           1'PRINTOUT ')
      252   DO 253 I=1,ITT
            WRITE(CRT,254)
      254   FORMAT(iX,'     STANDARD NUMBER : ',$)
            READ(CRT,1003)LENGTH,FNUM
            IF(LENGTH.LE.0)GO TO 10000
            NUM=FNUM
            WRITE(LUN'IVAR)NUM
      253   CONTINUE
            GO TO K
      450   IVAR=1
            IVAR=2
            ASSIGN 450 TO K
```

```
C
C  THIS SECTION ENABLES MODIFICATION OF AN EXISTING FILE
500   WRITE(CRT,501)
501   FORMAT(1X,'WHICH PARAMETER DO YOU WISH TO MODIFY ? ',$)
      READ(CRT,1000)LENGTH,ANS
      IF(LENGTH.LE.0)GO TO 10000
      IF(ANS.EQ.UN)GO TO 600
      IF(ANS.EQ.GN)GO TO 610
      IF(ANS.EQ.TH)GO TO 620
      IF(ANS.EQ.BG)GO TO 630
      IF(ANS.EQ.BS)GO TO 640
      IF(ANS.EQ.BT)GO TO 650
      IF(ANS.EQ.CT)GO TO 660
      IF(ANS.EQ.VR)GO TO 670
      IF(ANS.EQ.DT)GO TO 680
      IF(ANS.EQ.MF)GO TO 690
      IF(ANS.EQ.BA)GO TO 700
      IF(ANS.EQ.WD)GO TO 720
      IF(ANS.EQ.NS)GO TO 730
      IF(ANS.EQ.F0)GO TO 740
      IF(ANS.EQ.WF)GO TO 750
      IF(ANS.EQ.RC)GO TO 760
      IF(ANS.EQ.BR)GO TO 770
      IF(ANS.EQ.BP)GO TO 780
      IF(ANS.EQ.RO) GOTO 782
      IF(ANS.EQ.RS) GOTO 784
      IF(ANS.EQ.OR)GO TO 790
      GO TO 500
600   DO 601 I=1,32,4
601   READ(LUN'IVAR)A(I),A(I+1),A(I+2),A(I+3)
      WRITE(CRT,51)
      WRITE(CRT,1004)(A(I),I=1,32)
      IF(EDT.EQ.1)GO TO 2000
      IVAR=IVAR-8
      GO TO 52
610   IVAR=IVAR+8
      READ(LUN'IVAR)FNUM
      WRITE(CRT,61)
      WRITE(CRT,1007)FNUM
      IF(EDT.EQ.1)GO TO 2000
      IVAR=IVAR-1
      GO TO 62
      C
0333  620    IVAR=IVAR+9
0334         READ(LUN'IVAR)FNUM
0335         WRITE(CRT,71)
0336         WRITE(CRT,1007)FNUM
0337         IF(EDT.EQ.1)GO TO 2000
0339         IVAR=IVAR-1
0340         GO TO 72
0341  630    IVAR=IVAR+10
0342         READ(LUN'IVAR)NUM
0343         WRITE(CRT,81)
0344         ANS=NO
0345         IF(NUM.EQ.1)ANS=YES
0347         WRITE(CRT,1006)ANS
0348         IF(EDT.EQ.1)GO TO 2000
0350         IVAR=IVAR-1
0351         GO TO 82
0352  640    IVAR=IVAR+11
0353         READ(LUN'IVAR)NUM
0354         WRITE(CRT,91)
0355         ANS=NO
0356         IF(NUM.EQ.1)ANS=YES
0358         WRITE(CRT,1006)ANS
0359         IF(EDT.EQ.1)GO TO 2000
0361         IVAR=IVAR-1
0362         GO TO 92
0363  650    IVAR=IVAR+12
0364         READ(LUN'IVAR)FNUM
0365         WRITE(CRT,101)
0366         WRITE(CRT,1007)FNUM
0367         IF(EDT.EQ.1)GO TO 2000
0369         IVAR=IVAR-1
0370         GO TO 102
0371  660    IVAR=IVAR+13
0372         READ(LUN'IVAR)FNUM
0373         WRITE(CRT,111)
0374         WRITE(CRT,1007)FNUM
0375         IF(EDT.EQ.1)GO TO 2000
0377         IVAR=IVAR-1
0378         GO TO 112
0379  670    IVAR=IVAR+14
0380         READ(LUN'IVAR)FNUM
0381         WRITE(CRT,121)
0382         WRITE(CRT,1007)FNUM
0383         IF(EDT.EQ.1)GO TO 2000
0385         IVAR=IVAR-1
0386         GO TO 122
```

```
C
680     IVAR=IVAR+15
        READ(LUN'IVAR)FNUM
        WRITE(CRT,131)
        WRITE(CRT,1007)FNUM
        IF(EDT.EQ.1)GO TO 2000
        IVAR=IVAR-1
        GO TO 132
690     IVAR=IVAR+16
        READ(LUN'IVAR)FNUM
        WRITE(CRT,141)
        WRITE(CRT,1007)FNUM
        IF(EDT.EQ.1)GO TO 2000
        IVAR=IVAR-1
        GO TO 142
700     IVAR=IVAR+17
        DO 701 I=1,32,4
701     READ(LUN'IVAR)A(I),A(I+1),A(I+2),A(I+3)
        WRITE(CRT,151)
        WRITE(CRT,1004)(A(I),I=1,32)
        IF(EDT.EQ.1)GO TO 2000
        IVAR=IVAR-8
        GO TO 152
710     IVAR=IVAR+25
        READ(LUN'IVAR)NUM
        WRITE(CRT,171)
        ANS=NO
        IF(NUM.EQ.1)ANS=YES
        WRITE(CRT,1006)ANS
        IF(EDT.EQ.1)GO TO 2000
        IVAR=IVAR-1
        GO TO 172
720     IVAR=IVAR+26
        READ(LUN'IVAR)NUM
        WRITE(CRT,181)
        WRITE(CRT,1005)NUM
        IF(EDT.EQ.1)GO TO 2000
        IVAR=IVAR-1
        GO TO 182
730     IVAR=IVAR+27
        READ(LUN'IVAR)NUM
        WRITE(CRT,191)
        WRITE(CRT,1005)NUM
        IF(EDT.EQ.1)GO TO 2000
        IVAR=IVAR-1
        GO TO 192
```

```
C
740   IVAR=IVAR+28
      READ(LUN'IVAR)FNUM
      WRITE(CRT,201)
      WRITE(CRT,1007)FNUM
      IF(EDT.EQ.1)GO TO 2000
      IVAR=IVAR-1
      GO TO 202
750   IVAR=IVAR+29
      READ(LUN'IVAR)NUM
      WRITE(CRT,211)
      WRITE(CRT,1005)NUM
      IF(EDT.EQ.1)GO TO 2000
      IVAR=IVAR-1
      GO TO 212
760   IVAR=IVAR+30
      READ(LUN'IVAR)NUM
      WRITE(CRT,221)
      ANS=NO
      IF(NUM.EQ.1)ANS=YES
      WRITE(CRT,1006)ANS
      IF(EDT.EQ.1)GO TO 2000
      IVAR=IVAR-1
      GO TO 222
770   IVAR=IVAR+31
      READ(LUN'IVAR)NUM
      WRITE(CRT,231)
      WRITE(CRT,1005)NUM
      IF(EDT.EQ.1)GO TO 2000
      IVAR=IVAR-1
      GO TO 232
780   IVAR=IVAR+32
      READ(LUN'IVAR)NUM
      WRITE(CRT,241)
      WRITE(CRT,1005)NUM
      IF(EDT.EQ.1)GO TO 2000
      IVAR=IVAR-1
      GO TO 242
```

```
C
782    IVAR=IVAR+33
       READ(LUN'IVAR)NUM
       WRITE(CRT,244)
       ANS=NO
       IF(NUM.EQ.1) ANS=YES
       WRITE(CRT,1006)ANS
       IF(EDT.EQ.1) GOTO 2000
       IVAR=IVAR-1
       GOTO 245
784    IVAR=IVAR+34
       DO 785 I=1,32,4
785    READ(LUN'IVAR)A(I),A(I+1),A(I+2),A(I+3)
       WRITE(CRT,247)
       WRITE(CRT,1004)(A(I),I=1,32)
       IF(EDT.EQ.1) GOTO 2000
       IVAR=IVAR-8
       GOTO 248
790    IVAR=IVAR+27
       READ(LUN'IVAR)NUM1
       ITT=NUM1
       WRITE(CRT,251)
       IVAR=IVAR+14
796    WRITE(CRT,792)
792    FORMAT(1X,'       : ',$)
795    DO 793 I=1,ITT
       READ(LUN'IVAR)NUM
       WRITE(CRT,794)NUM
794    FORMAT('+',I3,$)
       IF(I.LT.18)GO TO 793
       IF(ITT.EQ.18)GO TO 793
       ITT=ITT-18
       GO TO 796
793    CONTINUE
       IF(EDT.EQ.1)GO TO 2000
       ITT=NUM1
       IVAR=IVAR-NUM1
       WRITE(CRT,797)
797    FORMAT(' ')
       GO TO 252
```

```
C
900   CONTINUE
      CRT=6
      EDT=1
      WRITE(CRT,901)
901   FORMAT('1','      PREGA UNKNOWN SAMPLE PARAMETERS',//)
      II=0
2000  II=II+1
      IVAR=2
      IF(II.GT.21)GO TO 2001
      GO TO (600,610,620,630,640,650,660,670,680,690,700,720,730,740
     1,750,760,770,780,782,784,790) II
2001  CONTINUE
      EDT=0
      WRITE(CRT,1009)
      WRITE(CRT,1008)
      WRITE(CRT,1008)
      WRITE(CRT,1008)
      WRITE(CRT,1008)
      WRITE(CRT,1008)
      WRITE(CRT,1008)
      WRITE(CRT,1008)
      CRT=5
      GO TO 1500
1000  FORMAT(Q,A2)
1001  FORMAT(Q,32A1)
1002  FORMAT(Q,1A1)
1003  FORMAT(Q,E12.0)
1004  FORMAT(1X ,'      : ',32A1,/)
1005  FORMAT('+',': ',I5,/)
1006  FORMAT('+',': ',1A1,/)
1007  FORMAT('+',': ',G14.6,/)
1008  FORMAT('+','                                                    ')
1009  FORMAT(' ')
11000 CONTINUE
      END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

A        000006  LOGICAL*1 ARRAY (32)
CC       000126  INTEGER*2 VARIABLE
EDT      003236  INTEGER*2 VARIABLE
UN       000046  INTEGER*2 VARIABLE
OP       000050  INTEGER*2 VARIABLE
CB       000052  INTEGER*2 VARIABLE
BG       000054  INTEGER*2 VARIABLE
BS       000056  INTEGER*2 VARIABLE
BT       000060  INTEGER*2 VARIABLE
UT       000122  INTEGER*2 VARIABLE
VR       000062  INTEGER*2 VARIABLE
DT       000064  INTEGER*2 VARIABLE
MF       000066  INTEGER*2 VARIABLE
SB       000070  INTEGER*2 VARIABLE
WD       000072  INTEGER*2 VARIABLE
NS       000074  INTEGER*2 VARIABLE
F0       000076  INTEGER*2 VARIABLE
WF       000100  INTEGER*2 VARIABLE
RC       000102  INTEGER*2 VARIABLE
BR       000104  INTEGER*2 VARIABLE
BP       000106  INTEGER*2 VARIABLE
ANS      003240  INTEGER*2 VARIABLE
YES      000132  INTEGER*2 VARIABLE
CRT      003242  INTEGER*2 VARIABLE
RO       000110  INTEGER*2 VARIABLE
RS       000112  INTEGER*2 VARIABLE
GN       000114  INTEGER*2 VARIABLE
TH       000116  INTEGER*2 VARIABLE
OR       000120  INTEGER*2 VARIABLE
CT       000136  INTEGER*2 VARIABLE
BA       000140  INTEGER*2 VARIABLE
NN       000124  INTEGER*2 VARIABLE
LL       000130  INTEGER*2 VARIABLE
NO       000134  INTEGER*2 VARIABLE
LUN      003244  INTEGER*2 VARIABLE
IVAR     003246  INTEGER*2 VARIABLE
M        003250  INTEGER*2 VARIABLE
LENGTH   003252  INTEGER*2 VARIABLE
K        003254  INTEGER*2 VARIABLE
I        003256  INTEGER*2 VARIABLE
FNUM     003260  REAL*4    VARIABLE
NUM      003264  INTEGER*2 VARIABLE
ITT      003266  INTEGER*2 VARIABLE
NUM1     003270  INTEGER*2 VARIABLE
II       003272  INTEGER*2 VARIABLE
```

```
C     NAI    MOD 44-0434-02    FLPY M12    31 AUG 77    L. HOLMES(SB)
C     MODIFIED 5 JUN 81 BY G. PHILLIPS, NRL
C
C
C       MODIFIED TO "PREGA" JULY 82 BY B. G. GLAGOLA, NRL
C
C
C       COPYRIGHT (C) 1977 BY NUCLEAR DATA INCORPORATED
C       SCHAUMBURG, ILLINOIS 60196
C
C
C       THIS DOCUMENTATION IS FURNISHED INCLUDING THE COPYRIGHT
C       NUCLEAR DATA INCORPORATED. IT MUST NOT BE REMOVED AND MAY NOT
C       BE REPRODUCED OR REDISTRIBUTED WITHOUT THE WRITTEN EXPLICIT
C       PERMISSION OF NUCLEAR DATA.
C
C
C     PURPOSE OF THE PROGRAM:
C     THIS PROGRAM DETERMINES THE ACTIVITY OF EACH NUCLIDE IN A SPECTRUM
C     CONTAINING SEVERAL NUCLIDES WHOSE IDENTITIES ARE KNOWN BUT WHOSE
C     ACTIVITIES ARE UNKNOWN. THE PROGRAM ASSUMES SPECTRAL DATA HAS
C     BEEN ACQUIRED VIA A SODIUM IODIDE DETECTOR.
C
C
C     METHOD:
C     THE PROGRAM MAKES A CHANNEL BY CHANNEL COMPARISON BETWEEN THE
C     UNKNOWN SPECTRUM AND A SERIES OF STANDARD SPECTRA. IT DETERMINES
C     ACTIVITIES BY LEAST SQUARES RESOLUTION OF THE SAMPLE DATA AND
C     USES PIVOTAL REGRESSION ANALYSIS TECHNIQUES TO CALCULATE AND
C     LIST THE ACTIVITY OF EACH NUCLIDE WITH ESTIMATED ERROR
C     AND OTHER INFORMATION ABOUT THE ANALYSIS.
C
C
C
C
```

```
C
C
C     VARIABLE DEFINITIONS
C
C     GENERAL CONTROL PARAMETERS
C     (USER DEFINED)
C     NS          =  NUMBER OF ISOTOPES IN STANDARD LIBRARY (MAX=80)
C     M           =  NUMBER OF CHANNELS PER STANDARD (MAX=512)
C     NIT         =  NUMBER OF ITERATION FOR THRESHOLD & GAIN CALC.
C     NBA         =  BACKGROUND SUBTRACT (1=YES,0=NO)
C     NZ          =  INITIAL CHANNEL FOR COMPUTATION
C     MF          =  FINAL CHANNEL FOR COMPUTATION
C     TB          =  COUNTING TIME FOR BACKGROUND
C     Q           =  REJECTION COEFFICIENT
C     ASC(I)      =  STANDARD BACKGROUND SPECTRUM (FILE.ELEMENT(,DEVICE))
C
C
C
C
C
C     CONTROL PARAMETERS FOR EACH LIBRARY ELEMENT
C
C     TISOT(JJ) =  ISOTOPE NAME (8 CHARACTERS MAX)
C     HA(I)       =  HALF-LIFE (SECS.)
C     TST(I)      =  COUNTINUE TIME OF STANDARD
C     AC(I)       =  ACTIVITY
C     ASC(I)      =  STANDARD SPECTRUM (FILE.ELEMENT(,DEVICE))
C
C
```

```
C
C
C     CONTROL PARAMETERS FOR UNKNOWN SAMPLE
C     (USER DEFINED)
C     NBR         =  BACKGROUND SUPPLIED ? (YES=1,NO=0)
C     NBS         =  BACKGROUND SUBTRACT ? (YES=1,NO=0)
C     TB          =  COUNTING TIME FOR BACKGROUND (SECS)
C     TSA         =  COUNTING TIME FOR UNKNOWN SAMPLE (SECS)
C     VRED        =  VOLUME REDUCTION FACTOR (INVERSE OF DILUTION FACTOR)
C     DAY         =  DECAY TIME (SECS)
C     VM          =  RESULT MULTIPLICATION FACTOR
C     ASC(JJ)     =  SAMPLE & BACKGROUND SPECTRUM (FILE.ELEMENT(,DEVICE))
C
C
C
C
C     UNKNOWN SAMPLE OPTION VARIABLES
C     (USER DEFINED)
C     NW          =  WEIGHTING FACTOR DETERMINATION (BASED ON ACTUAL
C                    COUNTS/CHANNEL=0, BASED ON CALCULATED COUNTS/CHN=1)
C     N           =  NUMBER OF LIBRARY SPECTRA TO BE FITTED TO SAMPLE
C     F0          =  F-TEST REJECTION COEFFICIENT
C     LW          =  WEIGHTING FACTOR CALCULATION ((COUNTS)**-1=-1,
C                 =  (SIGMA)**-2=0, 1=1)
C     NEWST       =  APPLY REJECTION COEFFICIENT ? (YES=1,NO=0)
C     BR          =  BACKGROUND REGRESSION (TRIAL SET, CHOSEN STANDARDS=0,
C                   BACKGROUND ONLY=1)
C     BP          =  PIVOT ON BACKGROUND ALLOWED ? (0=NO,1=YES)
C     IS(I)       =  NUMBER OF LIBRARY STANDARD USED DURING BUILDUP REGRESSION
C     ISL(I)      =  NUMBER OF LIBRARY STANDARD TO BE FIT IN ORDER OF
C                    DESIRED PRINTOUT
C
```

```
      C
0001        COMMON/X/FUD
0002        DIMENSION A(10,10),Y(513),Z(10),S(512)    ,STD(10),B(10),
           1 R(512),W(512),IR(512),MY(10),FPM(10),FP(10),
           2 BA(512),SS(10), AC(10),HA(10),IS(10), TST(10),HAT(10),
           3 TNAME(36),TISOT(40),TISO(20),IT(10),SNAM(2),
           4F R(10,64),HH(6),SIR(50),ISL(10),HF(4),
           5IFORN(20),IFORKL(30),CC(10,10)
0003        DOUBLE PRECISION HG,HR,HT,HF
0004        DATA HF/8HFULL SET,8HTEST SET,8HBACKWARD,8HFORWARD /
0005        DATA HG/8HREGRESN./,HR/8HRESIDUAL/,HT/8HTOTAL   /
0006        DATA HH/4HSOUR,4HCE  ,4H  DF,4H  SS,4H  MS,4H  F /
0007        EQUIVALENCE (BA,IR),(ARR,S),(TNAME(7),SNAM(1))
0008        LOGICAL*1 ASC(60),DEF(10),EF(10)
0009        LOGICAL*1 TNAME
0010        REAL TISO
0011        INTEGER C,D,E,F,MY,P,BR,BP
0012        DATA E/0/
0013        DATA MY/1,2,4,8,16,32,64,128,256,512/
0014        DATA DEF/1HD,1HE,1HF,1H ,1H ,1H8,1H,,1H ,1H ,1H /
0015        DATA TIS1/4HSAMP/,TIS2/4HLE  /
0016        DEF(9)=.FALSE.
0017        DEF(10)=.FALSE.
0018        ASC(33)=.FALSE.
0019        ASC(34)=.FALSE.
0020        INTEGER FILE
0021        FILE=8
      C
      C
0022        DEFINE FILE 9(4096,2,U,IDAT)
0023        DEFINE FILE 11(4096,2,U,IDAT)
0024        DEFINE FILE   10(0,2,U,IVAR)
```

```
      MI=9
      MIU=11
      MO=6
      WRITE(MO,900)
  900 FORMAT('1','     OUTPUT FOR PREGA ANALYSIS PROGRAM',///)
C     CHANGE RANGE OF NEXT DO LOOP TO MAX NR OF CHANNELS
      DO  1  I=1,512
    1 BA(I)=0.0
      NNN=0
      IPO=0
C
C     GENERAL CONTROL PARAMETERS
C
    2 READ(MI'1)NS
      READ(MI'2)M
      READ(MI'3)NIT
      READ(MI'4)NBA
      READ(MI'5)NZ
      READ(MI'6)MF
      READ(MI'40)TB
      READ(MI'41)Q
      NCH=MF-NZ+1
C
C     (USER DEFINED)
C     NS          =  NUMBER OF ISOTOPES IN STANDARD LIBRARY (MAX=80)
C     M           =  NUMBER OF CHANNELS PER STANDARD (MAX=512)
C     NIT         =  NUMBER OF ITERATION FOR THRESHOLD & GAIN CALC.
C  NIT IS TO BE REMOVED IN FUTURE STREAMLINING OF PROGRAM
C     NBA         =  BACKGROUND SUBTRACT (1=YES,0=NO)
C     NZ          =  INITIAL CHANNEL FOR COMPUTATION
C     MF          =  FINAL CHANNEL FOR COMPUTATION
C     TB          =  COUNTING TIME FOR BACKGROUND
C     Q           =  REJECTION COEFFICIENT
C
C
C     ****************************************************************
D     WRITE               (MO, 901)NS,M,NIT,NBA,NZ,MF
      IF(NS)897,897,3
C     ****************************************************************
C
C
C
    3 JJ=1
      DO 10 I=1,NS
      IDAT=65+(I-1)*16
```

```
C
C
C     CONTROL PARAMETERS FOR EACH LIBRARY ELEMENT
C
      READ(MI'IDAT)TISOT(JJ)
      JJ=JJ+1
      READ(MI'IDAT)TISOT(JJ)
      JJ=JJ+1
      IDAT=IDAT+8
      READ(MI'IDAT)HA(I)
      READ(MI'IDAT)TST(I)
      READ(MI'IDAT)AC(I)
C
C     TISOT(JJ) =  ISOTOPE NAME (8 CHARACTERS MAX)
C     HA(I)     =  HALF-LIFE (SECS.)
C     TST(I)    =  COUNTINUE TIME OF STANDARD
C     AC(I)     =  ACTIVITY
C
   10 CONTINUE
```

```
      IF (NBA) 50,50,40
   40 IDAT=42
      DO 45 J=1,32,4
C
C     ASC(I)    =  STANDARD BACKGROUND SPECTRUM (FILE.ELEMENT(,DEVICE))
C
C
   45 READ(MI'IDAT)ASC(J),ASC(J+1),ASC(J+2),ASC(J+3)
      WRITE(MO,902)(ASC(I),I=1,32)
  902 FORMAT(1X'STANDARD BACKGROUND: ',32A1)
      CALL       GET(ASC,BA,IVAR,FILE,M)
      WRITE              (MO,903)(BA(I),I=1,M)
C
   50 DO 130  I=1,NS
      IDAT=65+(I-1)*16+2
      DO 52 J=1,32,4
C
C
C     ASC(I)    =  STANDARD SPECTRUM (FILE.ELEMENT(,DEVICE))
C
   52 READ(MI'IDAT)ASC(J),ASC(J+1),ASC(J+2),ASC(J+3)
      CALL       GET(ASC,S,IVAR,FILE,M)
C
C     ******************************************************************
         IF(NBA) 70,70,60
C
C
C     SUBTRACT BACKGROUND (MULTIPLIED BY COUNT TIME OF STANDARD/ COUNT
C     TIME OF BACKGROUND) FROM STANDARD
C
   60 FAT=TST(I)/TB
      DO 65  J=1,M
      S(J)=S(J)-BA(J)*FAT
   65 CONTINUE
C
```

```
      C
      C     DON'T LET STANDARD COUNTS BE LESS THAN .01
      C
0074     70 DO 75 J=1,M
0075        IF(S(J).LT.0.0)S(J)=.01
0077     75 CONTINUE
0078           IF(NBA) 100,100,80
0079     80 WRITE(MO,904)(ASC(II),II=1,32)
0080    904 FORMAT('0','BACKGROUND SUBTRACTED STANDARD: ',32A1)
0081        WRITE(MO,903)(S(J),J=1,M)
0082    100 CONTINUE
      C
      C
      C     SAVE STANDARD SPECTRUM IN VIRTUAL MEMORY OUT ON DISK
      C
0083        IVAR=(I-1)*M+1
0084        DO 120 K=1,M
0085    120 WRITE(10'IVAR)S(K)
0086    130 CONTINUE
0087    170 WRITE            (MO,908)
      C     ******************************************************************
      C
      C
      C      CONTROL PARAMETERS FOR UNKNOWN SAMPLE
      C
0088         IDAT=12
0089         READ(MIU'IDAT)NBR
0090         READ(MIU'IDAT)NBS
0091         READ(MIU'IDAT)TB
0092         READ(MIU'IDAT)TSA
0093         READ(MIU'IDAT)VRED
0094         READ(MIU'IDAT)DAY
0095         READ(MIU'IDAT)VM
      C
      C      (USER DEFINED)
      C      NBR        =  BACKGROUND SUPPLIED ? (YES=1,NO=0)
      C      NBS        =  BACKGROUND SUBTRACT ? (YES=1,NO=0)
      C      TB         =  COUNTING TIME FOR BACKGROUND (SECS)
      C      TSA        =  COUNTING TIME FOR UNKNOWN SAMPLE (SECS)
      C      VRED       =  VOLUME REDUCTION FACTOR (INVERSE OF DILUTION FACTOR)
      C      DAY        =  DECAY TIME (SECS)
      C      VM         =  RESULT MULTIPLICATION FACTOR
      C
      C
      C      ******************************************************************
```

```
  180 IF (VM) 190,190,200
  190 VM=1.0
  200 IF (VRED) 210,210,220
  210 VRED=1.0
  220 IF (TB) 230,230,240
  230 FS=0.0
      GO TO 250
  240 FS=TSA/TB
  250 FX=FS+FS**2
      IF(NBR) 270,270,260
  260 CONTINUE
C     ******************************************************************
C
C
C     ASC(JJ)    =  SAMPLE & BACKGROUND SPECTRUM (FILE.ELEMENT(,DEVICE))
C
      DO 265 JJ=1,32,4
  265 READ(MIU'IDAT)ASC(JJ),ASC(JJ+1),ASC(JJ+2),ASC(JJ+3)
C
C      WRITE(MO,910)(ASC(I),I=1,32)
C910   FORMAT(1H1,'SAMPLE BACKGROUND: ',32A1)
C
      CALL       GET(ASC,BA,IVAR,FILE,M)
C      WRITE            (MO,903)(BA(I),I=1,M)
  270 CONTINUE
      IDAT=2
C
      DO 280 JJ=1,32,4
  280 READ(MIU'IDAT)ASC(JJ),ASC(JJ+1),ASC(JJ+2),ASC(JJ+3)
```

```
C
      WRITE(MO,911)(ASC(I),I=1,32)
  911 FORMAT(' ','SAMPLE SPECTRUM: ',32A1)
C
C
      DO 290 I=1,32
  290 TNAME(I)=ASC(I)
      CALL       GET(ASC,Y,IVAR,FILE,M)
C
C     ******************************************************************
      IF (NBS) 310,310,300
  300 DO 305 I=1,M
  305 Y(I)=Y(I)-BA(I)*FS
  310 SB=0.0
      DO 320 I=NZ,MF
  320 SB=SB+BA(I)
      WRITE            (MO,903)(Y(I), I=1,M)
      S1=0.0
      S2=0.0
      DO 330 I=NZ,MF
  330 S1=S1+Y(I)
      S2=S1+SB*FX
      WRITE            (MO, 912)SB,S1
C
C
C     SAVE ORIGINAL SAMPLE SPECTRUM OUT ON DISK
C
      IVAR=(NS+2)*M+1
      DO 340 I=1,M
  340 WRITE(10'IVAR)Y(I)
C
```

```
      C
      C
0135        IDAT=IDAT+18
      C
      C     UNKNOWN SAMPLE OPTION VARIABLES
      C
0136        NB=0
0137        READ(MIU'IDAT)NW
0138        READ(MIU'IDAT)N
0139        READ(MIU'IDAT)F0
0140        READ(MIU'IDAT)LW
0141        READ(MIU'IDAT)NEWST
0142        READ(MIU'IDAT)BR
0143        READ(MIU'IDAT)BP
0144        READ(MIU'IDAT)KRO
0145        DO 350 I=1,32,4
0146    350 READ(MIU'IDAT) ASC(I),ASC(I+1),ASC(I+2),ASC(I+3)
0147        NDET=0
      C
      C     (USER DEFINED)
      C     NW          =  WEIGHTING FACTOR DETERMINATION (BASED ON ACTUAL
      C                    COUNTS/CHANNEL=0, BASED ON CALCULATED COUNTS/CHN=1)
      C     N           =  NUMBER OF LIBRARY SPECTRA TO BE FITTED TO SAMPLE
      C     F0          =  F-TEST REJECTION COEFFICIENT
      C     LW          =  WEIGHTING FACTOR CALCULATION ((COUNTS)**-1=-1,
      C                 =  (SIGMA)**-2=0, 1=1)
      C     NEWST       =  APPLY REJECTION COEFFICIENT ? (YES=1,NO=0)
      C     BR          =  BACKGROUND REGRESSION? (TRIAL SET, CHOSEN SET
      C                    OF STANDARDS=0, BACKGROUND ONLY=0)
      C
      C     BP          =  PIVOT ON BACKGROUND ALLOWED ?
      C                    =0 NO PIVOT ON BACKGROUND ALLOWED
      C                    =1 ALLOW PIVOT ON BACKGROUND
      C     KRO =  OUTPUT RESIDUALS ? (YES=1,NO=0)
      C     ASC =  FILENAME FOR RESIDUALS
      C     IS(I)       =  NUMBER OF LIBRARY STANDARD TO BE FIT IN ORDER OF
      C                    DESIRED PRINTOUT
```

```
C
      DO 360 I=1,NS
  360 ISL(I)=0
      DO 370 I=1,N
      READ(MIU'IDAT)IS(I)
  370 ISL(I)=IS(I)
C
      DO 380 J=1,N
      DO 375 I=1,NS
      IF(IS(J)-I) 372,372,375
  372 II=2*I-1
C
C
C     STORE IN ARRAY TISO ONLY THOSE LIBRARY NAMES ACTUALLY USED IN THE
C     LEAST SQUARES FIT
C
      JJ=2*J-1
      TISO(JJ)=TISOT(II)
      II=II+1
      JJ=JJ+1
      TISO(JJ)=TISOT(II)
C
C     ***************************************************************************
      GO TO 380
  375 CONTINUE
  380 CONTINUE
      TISO((N+1)*2-1)=SNAM(1)
      TISO((N+1)*2)=SNAM(2)
      IS(N+1)=N+1
      LN=N+1
      DO 390 J=1,N
  390 IT(J)=J
      DO 395 I=1,N
      IF(IS(I).EQ.1) GO TO 400
  395 CONTINUE
      WRITE(MO,913)
  913 FORMAT('0',5X,45('*')/5X,'*',45X,'*'/5X,'*',4X,
     X'BACKGROUND NOT INCLUDED AS A STANDARD',4X,'*'/
     X5X,'*',45X,'*'/5X,45('*')/)
  400 CONTINUE
      IF(BP.EQ.0) GO TO 401
      WRITE(MO,936)
  401 DO 402 I=1,N
  402 EF(I)=.FALSE.
C
C     SET UP ARRAY A(N+1,N+1)
C
  405 DO 420 I=1,M
      IF(LW) 422,412,410
  410 W(I)=1.0
      GO TO 420
  412 T=Y(I)+BA(I)*FX
      IF (T- 1.0) 414,414,416
  414 W(I)=1.0
      GO TO 420
  416 W(I)=1.0/T
  420 CONTINUE
      GO TO 424
  422 DO 423 I=1,M
  423 W(I)=1.0/(Y(I)+1.0)
C
```

```
   424 SUMW=0.0
       DO 425 I=1,M
   425 SUMW=SUMW+1.0/W(I)
       WRITE(MO,914)
   914 FORMAT('0','*****  CORRELATION MATRIX  *****'/)
       ENCODE(17,915,IFORN) LN
   915 FORMAT('(14X,',I2,'(2A4,2X)/)')
       WRITE(MO,IFORN) (TISO(I*2-1),TISO(I*2),I=1,LN)
       DO 446 L=1,N
       N5=IS(L)
       IVAR=(N5-1)*M+NZ
       DO 435 J=NZ,MF
  435  READ(10'IVAR)R(J)
C
C      COMPUTE WEIGHTED SUMS OF STANDARD SPECTRA
C
       SS(L)=0.
       DO 436 J=NZ,MF
436    SS(L)=SS(L)+R(J)*W(J)
       DO 444 K=L,N
       N6=IS(K)
       IVAR=(N6-1)*M+NZ
       DO 437 J=NZ,MF
  437  READ(10'IVAR)S(J)
442    SA=0.
```

```
C
C     GET LEAST SQUARES SUM COEFFICIENTS FOR UNKNOWN CONCENTRATIONS Z(I)
C     AND SAMPLE Y(I)
C
      DO 443 I=NZ,MF
  443 SA=SA+S(I)*R(I)*W(I)
      A(K,L)=SA
      A(L,K)=A(K,L)
  444 CONTINUE
      SA=0.
      DO 445 I=NZ,MF
      SA=SA+R(I)*Y(I)*W(I)
  445 CONTINUE
      B(L)=SA
      A(N+1,L)=SA
      A(L,N+1)=A(N+1,L)
  446 CONTINUE
      SA=0.
      DO 450 I=NZ,MF
      SA=SA+Y(I)*Y(I)*W(I)
  450 CONTINUE
      A(LN,LN)=SA
      ST=SA
C
C     COMPUTE WEIGHTED SUM OF UNKNOWN
C
      SS(LN)=0.
      DO 460 J=MZ,NF
460   SS(LN)=SS(LN)+Y(J)*W(J)
```

```
C
C          CALCULATE CORRELATION MATRIX
C
           DO 468 L=1,LN
           N5=IS(L)
           DO 465 K=L,LN
           N6=IS(K)
           SRD1=A(K,K)-SS(N6)**2/SUMW
           SRD2=A(L,L)-SS(N5)**2/SUMW
           CC(K,L)=(A(K,L)-SS(N6)*SS(N5)/SUMW)/SQRT(SRD1*SRD2)
    465    CONTINUE
           LSK=(L-1)*10+2
           KAC=LN-L+1
           ENCODE(25,916,IFORKL) LSK,KAC
    916    FORMAT('(2X,2A4,',I2,'X,',I2,'(F8.4,2X)/)')
           WRITE(MO,IFORKL) TISO(L*2-1),TISO(L*2),(CC(K,L),K=L,LN)
    468    CONTINUE
C
C          PRINT OUT WEIGHTED SUMS OF STANDARDS AND UNKNOWN
C
           WRITE(MO,905)
           WRITE(MO,906) (SS(I),I=1,LN)
```

```
      C
      C
      C
      C     DO PIVOTAL REGRESSION ANALYSIS
      C
      C
      C     REGRESSION WITH ALL REQUESTED STANDARDS
      C
0256        E=0
0257        P=N
0258        IF(BR.EQ.0) GO TO 479
0260        WRITE(MO,930)
0261        DO 470 I=1,N
0262        IF(ISL(I).EQ.1) GO TO 471
0264    470 CONTINUE
0265    471 N5=ISL(I)
0266        E=E+MY(N5)
0267        CALL PIVOT(A,I,EF,LN)
0268        IPO=1
0269        P=1
0270        GO TO 700
0271    479 DO 480 I=1,P
0272        E=E+MY(IS(I))
0273        CALL PIVOT(A,I,EF,LN)
0274    480 CONTINUE
0275        IPO=1
0276        GO TO 700
      C
```

```
C
C     SELECT TRIAL SUBSET OF MATRIX A
C
  500 F=E
      D=0
      IF(BR.EQ.0) GO TO 502
      F=0
      DO 501 I=1,N
      N5=ISL(I)
      F=F+MY(N5)
  501 CONTINUE
      GO TO 609
  502 P=N
  508 DO 510 I=1,N
      N5=ISL(I)
      IF(BP.EQ.1) GO TO 509
      IF(N5.EQ.1) GO TO 510
      IF(STD(I).LE.0.0) GO TO 511
  509 IF(Z(I)/STD(I).GE.1.0) GO TO 510
  511 CALL PIVOT(A,I,EF,LN)
      E=E-MY(N5)
      P=P-1
  510 CONTINUE
      IF(E.EQ.1.AND.Z(1).LT.0.0) GO TO 890
      IF(E.GT.0) GO TO 540
C
C     FIND LARGEST F-VALUE AND USE THAT STANDARD AS THE TRIAL SET
C
      FPS=0.0
      DO 530 I=1,N
      IF(Z(I)) 530,530,520
  520 IF(FP(I).LT.FPS) GO TO 530
      FPS=FP(I)
      J=I
  530 CONTINUE
      P=1
      E=E.AND.MY(ISL(J))
  540 IPO=2
      GO TO 700
```

```
C
C     BACKWARD PIVOT
C
  550 C=E
      IF(E.EQ.1.AND.BP.EQ.0) GO TO 800
  551 FMIN=10000.
      DO 555 I=1,N
      FPM(I)=1000.
      N5=ISL(I)
      IF(BP.EQ.1) GO TO 552
      IF(N5.EQ.1) GO TO 555
  552 IN=C.AND.MY(N5)
      IF(IN) 555,555,554
  554 FPM(I)=(NCH-P)*((A(I,LN)**2)/(-A(I,I)*A(LN,LN)))
      IF(A(I,LN).GT.0.0) FPM(I)=-FPM(I)
      IF(FPM(I).GE.FMIN) GO TO 555
      FMIN=FPM(I)
      J=I
  555 CONTINUE
      DF=NCH-P
      ZS=-A(J,LN)
      IF(FMIN.LT.0.0) FMIN=-FMIN
      CALL XQCALC(DF,FMIN,ZS,XF,QF,0)
      IF(XF.GE.F0) GO TO 560
      CALL PIVOT(A,J,EF,LN)
      E=E-MY(ISL(J))
      P=P-1
  560 IPO=3
      GO TO 700
  600 JDF=F-D
      IF(E.EQ.JDF) GO TO 800
```

```
C
C    FORWARD PIVOT
C
  609 IF(BR.EQ.1) C=E
      D=F-C
      IF(D.EQ.0) GO TO 630
      FMAX=0.0
      DF=NCH-P-1
      WRITE(MO,932)
      DO 620 I=1,N
      FPM(I)=0.0
      N5=ISL(I)
      IF(BP.EQ.1) GO TO 610
      IF(N5.EQ.1) GO TO 620
  610 IN=D.AND.MY(N5)
      IF(IN) 620,620,615
  615 INEG=0
      SQUC=A(I,I)*A(LN,LN)
      IF(SQUC.GT.0.0) GO TO 616
      SQUC=-1.0*SQUC
      INEG=1
  616 RHO=A(I,LN)/SQRT(SQUC)
      IF(INEG.EQ.1) RHO=-1.0*RHO
      R2=RHO**2
      IF(INEG.EQ.1) R2=-1.0*R2
      FPM(I)=DF*R2/(1.0-R2)
      ZS=A(I,LN)
      FPS=FPM(I)
      CALL XQCALC(DF,FPS,ZS,XF,QF,0)
       RMSR=A(LN,LN)*(1.0-R2)/DF
       WRITE(MO,933) TISO(I*2-1),TISO(I*2),FPS,QF,XF,RHO,RMSR
      IF(XF.LT.0.0) FPM(I)=-FPM(I)
      IF(FPM(I).LE.FMAX) GO TO 620
      FMAX=FPM(I)
      J=I
  620 CONTINUE
      ZS=A(J,LN)
      CALL XQCALC(DF,FMAX,ZS,XF,QF,0)
      IF(XF.LT.F0) GO TO 630
      CALL PIVOT(A,J,EF,LN)
      E=E+MY(ISL(J))
      P=P+1
      IPO=4
      BR=0
      GO TO 700
  630 IF(E.EQ.C) GO TO 800
      GO TO 550
```

```
C
C     OUTPUT SECTION FOR PARTIAL F,Q,X AND ANALYSIS OF VARIANCE
C
  700 J=0
      DF=NCH-P
      RSS=A(LN,LN)
      RMSR=RSS/DF
      DO 730 I=1,N
      N5=ISL(I)
      IN=E.AND.MY(N5)
      IF(IN) 730,730,720
  720 J=J+1
      TISOT(2*J-1)=TISO(2*I-1)
      TISOT(2*J)=TISO(2*I)
      Z(J)=-A(I,LN)*TST(N5)/TSA
      SQUC=A(I,I)
      IF(SQUC.LT.0.0) SQUC=-1.0*SQUC
      STD(J)=SQRT(SQUC*RMSR)*TST(N5)/TSA
      FP(J)=(Z(J)/STD(J))**2
  730 CONTINUE
      WRITE(MO,918) HF(IPO),SNAM(1),SNAM(2)
  918 FORMAT('1','RESULTS OF ',A8,' REGRESSION ON ',2A4,///)
      WRITE(MO,919) DF,RSS,RMSR
  919 FORMAT(' ','DEGREES OF FREEDOM=',F11.0/
     X13X,'RSS=',1PE11.3/13X,'MSR=',1PE11.3//)
      DO 740 I=1,P
      FPS=FP(I)
      ZS=Z(I)
      CALL XQCALC(DF,FPS,ZS,XF,QF,0)
      FP(I)=XF
      RMSR=A(LN,LN)*(1.0+FPS/DF)/(DF+1)
  738 WRITE(MO,920) TISOT(I*2-1),TISOT(I*2),Z(I),STD(I),FPS,QF,XF,RMSR
  920 FORMAT(' ',2X,2A4,1PE13.4,2X,'+-',1PE11.2,3X,'F=',
     X1PE12.4,3X,'Q=',1PE12.4,3X,'X=',1PE12.4,3X,'RMSR=',1PE12.4)
  740 CONTINUE
      GO TO (500,550,600,630),IPO
```

```
C
C    FINAL RESULT SECTION
C
  800 SG=0
      J=0
      IF(E.EQ.1.AND.Z(1).LT.0.0) GO TO 890
      DO 820 I=1,N
      N5=ISL(I)
      IN=E.AND.MY(N5)
      IF(IN) 820,820,815
  815 J=J+1
      SG=SG-A(I,LN)*B(I)
      TISOT(J*2-1)=TISO(I*2-1)
      TISOT(J*2)=TISO(I*2)
  820 CONTINUE
C
C
```

```
C
      CH=0.0
      VY=0.
      VU=0.0
      VVV=0.0
      JJJ=0
C
C
C       THIS LOOP READS FROM A SCRATCH FILE INTO CORE UP TO 15
C     STANDARDS, 64 CHANNELS AT A TIME SO THAT RESIDUALS AND STANDARD
C     ERRORS MAY BE DETERMINED
C
      DO 835 J=NZ,MF
      JJJ=JJJ+1
      SV=0.
      JJ=0
      DO 828 I=1,N
      N5=ISL(I)
      IN=E.AND.MY(N5)
      IF(IN) 828,828,822
  822 JJ=JJ+1
      IF(JJJ.GT.1.AND.JJJ.LT.65)GO TO 827
      JJJ=1
      LOOP=(MF+1)-J
      IF(LOOP.GT.64)LOOP=64
      IVAR=(N5-1)*M+J
      DO 826 II=1,LOOP
  826 READ(10'IVAR)ARR(JJ,II)
D     WRITE(MO,903)(ARR(JJ,II),II=1,LOOP)
  827 SV=SV+ARR(JJ,JJJ)*(-A(I,LN))
      IF(JJ.EQ.P) JJ=0
  828 CONTINUE
C
```

```
C
      RE=Y(J)-SV
      T=RE**2
      VY=VY+W(J)*T
      VU=VU+T
      TMO=ABS(0.1+SV)
      TMP=TMO+BA(J)*FX
      VVV=VVV+TMP
      IF (NW) 833,833,830
  830 IF (LW) 832,831,833
  831 W(J)=1.0/TMP
      GO TO 833
  832 W(J)=1.0/TMO
  833 RT=T/TMP
      CH=CH+RT
      TMP=SQRT(TMP)
C
C
C     CALCULATES RESIDUALS OVER STANDARD DEVIATION, R(J)
C
      R(J)=RE/TMP
      TE=Y(J)+BA(J)*FX
  835 CONTINUE
      SR=VY
      C
      C      CALCULATES NUMBER OF DEGREES OF FREEDOM, DN
      C
0500         DN=MF-P-NZ+1
0501         DR=DN
0502         DG=P
0503         DT=DR+DG
0504         CHDF=CH/DN
0505         VY=VY/DN
0506         WRITE(MO,896)
0507         WRITE(MO,934) SNAM(1),SNAM(2)
0508         WRITE             (MO, 921)CHDF
0509         J=0
0510     840 DO 844 I=1,N
0511         NS=ISL(I)
0512         IN=E.AND.MY(NS)
0513         IF(IN) 844,844,841
0514     841 J=J+1
      C
      C      DECAY CORRECTION DONE HERE
      C
0515         FD= EXP(0.693*DAY/HA(NS))
```

```
C
C    CORRECTED STANDARD ERROR DETERMINED FROM THE VARIANCE
C
       STD(J)=SQRT(-A(I,I)*A(LN,LN)/DN)*TST(N5)/TSA
       STD(J)=AC(  N5 )*FD*STD(J)*VM/VRED
C
C      RELATIVE CONCENTRATION IS CONVERTED TO ACTUAL CONCENTRATION LEVEL
C
       Z(J)=(-A(I,LN))*(TST(N5)/TSA)*AC(N5)*FD*VM/VRED
   844 CONTINUE
C
   846      WRITE           (MO,922)
       DO 847 I=1,P
       FPS=(Z(I)/STD(I))**2
       ZPS=Z(I)
       CALL XQCALC(DN,FPS,ZPS,XF,QF,0)
       RMSR=A(LN,LN)*(1.0+FPS/DR)/(DR+1)
       WRITE(MO,920) TISOT(I*2-1),TISOT(I*2),Z(I),STD(I),FPS,QF,XF,RMSR
   847 CONTINUE
C
C      WRITE ANALYSIS OF VARIANCE TABLE
C
       XG=SG/DG
       XR=SR/DR
       FF=XG/XR
       WRITE(MO,924)HH
       WRITE(MO,925)HG,DG,SG,XG,FF
       WRITE(MO,925)HR,DR,SR,XR
       WRITE(MO,925)HT,DT,ST
C
C      COMPUTE LLNL R-STATISTIC
C
       CALL RSTAT(NCH,P,R,NZ,MF)
```

```
C
C     COMPUTE FINAL STATISTICS FOR STANDARDS ELIMINATED
C     DURING PIVOTAL REGRESSION
C
      WRITE(MO,935)
      DF=NCH-P-1
      J=0
      DO 850 I=1,N
      N5=ISL(I)
      IN=D.AND.MY(N5)
      IF(IN) 850,850,848
  848 J=J+1
      TISOT(J*2-1)=TISO(I*2-1)
      TISOT(J*2)=TISO(I*2)
      ZS=A(I,LN)
      INEG=0
      SQUC=A(I,I)*A(LN,LN)
      IF(SQUC.GT.0.0) GO TO 849
      SQUC=-1.0*SQUC
      INEG=1
  849 RHO=A(I,LN)/SQRT(SQUC)
      IF(INEG.EQ.1) RHO=-1.0*RHO
      R2=RHO**2
      IF(INEG.EQ.1) R2=-1.0*R2
      FPS=DF*R2/(1.0-R2)
      CALL XQCALC(DN,FPS,ZS,XF,QF,0)
      RMSR=A(LN,LN)*(1.0-R2)/DF
      WRITE(MO,933) TISOT(J*2-1),TISOT(J*2),FPS,QF,XF,RHO,RMSR
  850 CONTINUE
       WRITE            (MO,926) SNAM(1),SNAM(2)
      DO 851  II=1,NZ-1
 851  R(II)=0.0
      WRITE            (MO,927)(R(J),J=1,MF)
853   K=0
      N2=MF-2
C
```

```
C
C     FIND SUSPICIOUS CHANNELS. THESE ARE DEFINED AS CHANNELS WHERE
C     STD/RESIDUAL IS GREATER THAN 10 OR WHERE STD/RESIDUAL IS GREATER
C     THAN 2 FOR THE CHANNELS ON EITHER SIDE OF THE ONE CONSIDERED.
C
      DO 870 J=NZ,N2
      T=ABS(R(J))
      IF(T-10.) 860,860,866
  860 IF(R(J-1)-2.0 )  870,862,862
  862 IF(R(J)-2.0 )  870,864,864
  864 IF(R(J+1)-2.0 ) 870,866,866
  866 K=K+1
      IR(K)=J
      SIR(K)=R(J)
      IF(K.GT.50) GO TO 872
  870 CONTINUE
  872 WRITE               (MO,928)
      WRITE               (MO,929)(IR(I),SIR(I),I=1,K)
      CALL RESID(R,NZ,MF)
      CALL RUNS(R,NZ,MF)
      IF(KRO.NE.1) GO TO 880
      DO 876 I=NZ,MF
      R(I)=R(I)*100+1000
      IF(R(I).GT.2000.) R(I)=2000.
      IF(R(I).LT.0.) R(I)=0.
 876  CONTINUE
      CALL PUT(ASC,R,IVAR,FILE,M)
 880  GO TO 895
```

```
      C
0597    901 FCRMAT(' ',1X,6I4)
0598    903  FORMAT  (2X,7F8.0)
0599    905 FORMAT('0',1X,'WEIGHTED SUMS OF STANDARDS AND UNKNOWN',/)
0600    906 FORMAT(1X,1PE11.4,1PE11.4,1PE11.4,1PE11.4,/)
0601    908 FORMAT(1H0)
0602    909 FORMAT(3A2,2X,  I2,I5,I3,I2,I10,  5E10.3)
0603    912 FORMAT(1H1,'BACKGD SUM=',1PE12.3,'      SAMPLE SUM=',1PE12.3,/)
0604    921 FORMAT(' ',1X,'FIT=',1PE11.3)
0605    922 FORMAT(1H0,'RESULTS=CONCENTRATIONS AND EST STANDARD ERRORS',/)
0606    926 FORMAT(1H1,' RATIO OF RESIDUALS OVER STD DEV PER CHANNEL FOR',
           X1X,2A4,/)
0607    927 FORMAT   (1X,F7.1, 6F8.1)
0608    928 FORMAT(1H0,'SUSPICIOUS CHANNELS',/)
0609    929 FORMAT   (5(I4,2X,F5.2),/)
0610    924 FORMAT(1H0,12X,'ANALYSIS OF VARIANCE'/1X,2A4,6X,A4,4X,3(A4,7X))
0611  925   FORMAT(1X,A8,F11.0,1P4E11.3)
0612    930 FORMAT(' ',14X,40('*')/15X,'*',8X,
           1'STARTING ANALYSIS WITH',8X,'*'/15X,'*',5X,
           2'TRIAL SET = BACKGROUND ONLY',6X,'*'/15X,
           340('*'))
0613    932 FORMAT(' ',///////////////////////////
           1' CALCULATED PARTIAL F-,Q- AND X-VALUES  TO ADD:')
0614    933 FORMAT(' ',2A4,3X,'F=',1PE12.4,3X,'Q=',1PE12.4,
           13X,'X=',1PE12.4,3X,'RHO=',1PE12.4,3X,'AMSR=',1PE12.4)
0615    934 FORMAT(' ',50('*')/13X,'FINAL RESULT FOR ',2A4,/50('*')//)
0616    935 FORMAT(' ',//////' FINAL STATISTICS FOR ELIMINATED STANDARDS')
0617    936 FORMAT(' ',10X,41('*')/11X,'*',5X,
           1'PIVOT ON BACKGROUND PERMITTED',5X,'*'/11X,41('*'))
0618    890 WRITE(MO,940)
0619        WRITE(7,940)
0620    940 FORMAT('  NONE OF THE CHOSEN STANDARDS RESULT'/
           X'IN A SIGNIFICANT CONTRIBUTION TO THE'/
           X'BACKGROUND SUBTRACTED SAMPLE SPECTRUM')
0621  895   WRITE(MO,896)
0622    896 FORMAT('1')
0623  897   CONTINUE
0624        END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

A        000006  REAL*4     ARRAY (10,10) VECTORED
Y        000626  REAL*4     ARRAY (513)
Z        004632  REAL*4     ARRAY (10)
S        004702  REAL*4     ARRAY (512)
STD      011702  REAL*4     ARRAY (10)
B        011752  REAL*4     ARRAY (10)
R        012022  REAL*4     ARRAY (512)
W        016022  REAL*4     ARRAY (512)
IR       022022  INTEGER*2  ARRAY (512)
MY       026022  INTEGER*2  ARRAY (10)
FPM      026046  REAL*4     ARRAY (10)
FP       026116  REAL*4     ARRAY (10)
BA       022022  REAL*4     ARRAY (512)
SS       026166  REAL*4     ARRAY (10)
AC       026236  REAL*4     ARRAY (10)
HA       026306  REAL*4     ARRAY (10)
IS       026356  INTEGER*2  ARRAY (10)
TST      026402  REAL*4     ARRAY (10)
HAT      026452  REAL*4     ARRAY (10)
TNAME    026522  LOGICAL*1  ARRAY (36)
TISOT    026566  REAL*4     ARRAY (40)
TISO     027026  REAL*4     ARRAY (20)
IT       027146  INTEGER*2  ARRAY (10)
SNAM     026530  REAL*4     ARRAY (2)
ARR      004702  REAL*4     ARRAY (10,64) VECTORED
HH       027172  REAL*4     ARRAY (6)
SIR      027222  REAL*4     ARRAY (50)
ISL      027532  INTEGER*2  ARRAY (10)
HF       027556  REAL*8     ARRAY (4)
IFORN    027616  INTEGER*2  ARRAY (20)
IFORKL   027666  INTEGER*2  ARRAY (30)
CC       027762  REAL*4     ARRAY (10,10) VECTORED
ASC      030602  LOGICAL*1  ARRAY (60)
DEF      030676  LOGICAL*1  ARRAY (10)
EF       030710  LOGICAL*1  ARRAY (10)
HG       030722  REAL*8     VARIABLE
HR       030732  REAL*8     VARIABLE
HT       030742  REAL*8     VARIABLE
C        034016  INTEGER*2  VARIABLE
D        034020  INTEGER*2  VARIABLE
E        030752  INTEGER*2  VARIABLE
F        034022  INTEGER*2  VARIABLE
P        034024  INTEGER*2  VARIABLE
BR       034026  INTEGER*2  VARIABLE
BP       034030  INTEGER*2  VARIABLE
FILE     034032  INTEGER*2  VARIABLE
TIS1     030754  REAL*4     VARIABLE
TIS2     030760  REAL*4     VARIABLE
IDAT     034034  INTEGER*2  VARIABLE
IVAR     034036  INTEGER*2  VARIABLE
MI       034040  INTEGER*2  VARIABLE
MIU      034042  INTEGER*2  VARIABLE
MO       034044  INTEGER*2  VARIABLE
I        034046  INTEGER*2  VARIABLE
NNN      034050  INTEGER*2  VARIABLE
IPO      034052  INTEGER*2  VARIABLE
NS       034054  INTEGER*2  VARIABLE
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

M        034056  INTEGER*2 VARIABLE
NIT      034060  INTEGER*2 VARIABLE
NBA      034062  INTEGER*2 VARIABLE
NZ       034064  INTEGER*2 VARIABLE
MF       034066  INTEGER*2 VARIABLE
TB       034070  REAL*4    VARIABLE
Q        034074  REAL*4    VARIABLE
NCH      034100  INTEGER*2 VARIABLE
JJ       034102  INTEGER*2 VARIABLE
J        034104  INTEGER*2 VARIABLE
GET      000000  REAL*4    PROCEDURE
FAT      034106  REAL*4    VARIABLE
II       034112  INTEGER*2 VARIABLE
K        034114  INTEGER*2 VARIABLE
NBR      034116  INTEGER*2 VARIABLE
NBS      034120  INTEGER*2 VARIABLE
TSA      034122  REAL*4    VARIABLE
VRED     034126  REAL*4    VARIABLE
DAY      034132  REAL*4    VARIABLE
VM       034136  REAL*4    VARIABLE
FS       034142  REAL*4    VARIABLE
FX       034146  REAL*4    VARIABLE
SB       034152  REAL*4    VARIABLE
S1       034156  REAL*4    VARIABLE
S2       034162  REAL*4    VARIABLE
NB       034166  INTEGER*2 VARIABLE
NW       034170  INTEGER*2 VARIABLE
N        034172  INTEGER*2 VARIABLE
F0       034174  REAL*4    VARIABLE
LW       034200  INTEGER*2 VARIABLE
NEWST    034202  INTEGER*2 VARIABLE
KRO      034204  INTEGER*2 VARIABLE
NDET     034206  INTEGER*2 VARIABLE
LN       034210  INTEGER*2 VARIABLE
T        034212  REAL*4    VARIABLE
SUMW     034216  REAL*4    VARIABLE
L        034222  INTEGER*2 VARIABLE
N5       034224  INTEGER*2 VARIABLE
N6       034226  INTEGER*2 VARIABLE
SA       034230  REAL*4    VARIABLE
ST       034234  REAL*4    VARIABLE
MZ       034240  INTEGER*2 VARIABLE
NF       034242  INTEGER*2 VARIABLE
SRD1     034244  REAL*4    VARIABLE
SRD2     034250  REAL*4    VARIABLE
SQRT     000000  REAL*4    PROCEDURE
LSK      034254  INTEGER*2 VARIABLE
KAC      034256  INTEGER*2 VARIABLE
PIVOT    000000  REAL*4    PROCEDURE
FPS      034260  REAL*4    VARIABLE
FMIN     034264  REAL*4    VARIABLE
IN       034270  INTEGER*2 VARIABLE
DF       034272  REAL*4    VARIABLE
ZS       034276  REAL*4    VARIABLE
XQCALC   000000  REAL*4    PROCEDURE
XF       034302  REAL*4    VARIABLE
QF       034306  REAL*4    VARIABLE
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

JDF      034312  INTEGER*2 VARIABLE
FMAX     034314  REAL*4    VARIABLE
INEG     034320  INTEGER*2 VARIABLE
SQUC     034322  REAL*4    VARIABLE
RHO      034326  REAL*4    VARIABLE
R2       034332  REAL*4    VARIABLE
RMSR     034336  REAL*4    VARIABLE
RSS      034342  REAL*4    VARIABLE
SG       034346  REAL*4    VARIABLE
CH       034352  REAL*4    VARIABLE
VY       034356  REAL*4    VARIABLE
VU       034362  REAL*4    VARIABLE
VVV      034366  REAL*4    VARIABLE
JJJ      034372  INTEGER*2 VARIABLE

V        034374  REAL*4    VARIABLE
LOOP     034400  INTEGER*2 VARIABLE
RE       034402  REAL*4    VARIABLE
TMO      034406  REAL*4    VARIABLE
ABS      000000  REAL*4    PROCEDURE
TMP      034412  REAL*4    VARIABLE
RT       034416  REAL*4    VARIABLE
TE       034422  REAL*4    VARIABLE
SR       034426  REAL*4    VARIABLE
DN       034432  REAL*4    VARIABLE
DR       034436  REAL*4    VARIABLE
DG       034442  REAL*4    VARIABLE
DT       034446  REAL*4    VARIABLE
CHDF     034452  REAL*4    VARIABLE
FD       034456  REAL*4    VARIABLE
EXP      000000  REAL*4    PROCEDURE
ZPS      034462  REAL*4    VARIABLE
XG       034466  REAL*4    VARIABLE
XR       034472  REAL*4    VARIABLE
FF       034476  REAL*4    VARIABLE
RSTAT    000000  REAL*4    PROCEDURE
N2       034502  INTEGER*2 VARIABLE
~~RESID    000000  REAL*4    PROCEDURE~~
RUNS     000000  REAL*4    PROCEDURE
PUT      000000  REAL*4    PROCEDURE

COMMON BLOCK /XV/        LENGTH 000004

FUD      000000  REAL*4    VARIABLE
```

```
ND 6600 LINKER      V02-A-1     LOAD MAP     5 OCT 1983  1:43:12 PM

SECTION ADDR   SIZE   ENTRY  ADDR   ENTRY  ADDR   ENTRY  ADDR
. ABS.  000000 000000 $LRECL 000210 $NLCHN 000006 $USRSW 000000
                      $RF1B3 000000
. ABS.  000000 000000 $SYSLB 000000
. ABS.  000000 000000 $TRACE 004737
. ABS.  000000 000000 $V004A 000001
.$$$$.& 020000 000000 .$$$$. 020000
X     & 020000 000004 X      020000
        020004 053706
        073712 001066 PIVOT  073712
        075000 001060 RSTAT  075000
        076060 000624 XQCALC 076060
        076704 000552 GET    076704
        077456 000614 PUT    077456
        100272 001364 RESID  100272
        101656 001342 RUNS   101656
        103220 000772 HISTO  103220
        104212 000106 CONCAT 104212
        104320 000060 NMI$II 104320 NMI$MI 104346 NMI$PI 104354
                      NPI$II 104364 NPI$MI 104370 NPI$PI 104374
        104400 000044 NMI$1P 104400 NPI$1I 104420 NPI$1M 104432
                      NPI$1P 104414
        104444 000142 SQRT   104444
        104606 000350 EXP    104606
        105156 000262 XFI$   105156 $PWRI  105156
        105440 000110 TAD$   105470 TAF$   105476 TAI$   105440
                      TAL$   105446 TAP$   105462 TAQ$   105454
        105550 000056 $OTIS  105550
        105626 000210 ISN$   105626 LSN$   105646 $ISNTR 105632
                      $LSNTR 105652
        106036 000034 END$   106036 ERR$   106054
        106072 000046 EOL$   106072
        106140 000044 RET$   106154 RET$F  106144 RET$I  106152
                      RET$L  106140
        106204 001106 IBR$   106212 IBW$   106204 $IBW   106216
        107312 000072 ENC$   107312
        107404 000020 IFR$   107404 IFW$   107416
        107424 001562 $FIO   110112
        111206 002344 DCO$   112632 ECO$   112624 FCO$   112620
                      GCO$   112612 ICI$   111214 ICO$   112366
                      OCI$   111206 OCO$   112360 RCI$   111410
                      $GET   111374
        113552 000110 $DUMPL 113552
        113662 000036 $GETFI 113662
        113720 000042 DII$IS 113732 DII$MS 113726 DII$PS 113720
                      DII$SS 113734 $DVI   113734
        113762 000040 MUI$IS 113774 MUI$MS 113770 MUI$PS 113762
                      MUI$SS 113776 $MLI   113776
        114022 000130 CIC$   114022 CID$   114022 CIF$   114032
                      CIL$   114144 CLC$   114022 CLD$   114022
                      CLF$   114032 CLI$   114150 $DI    114022
                      $RI    114032
```

```
114152 000120 $FCHNL 114152
114272 000044 BEQ$   1!4312 BGE$   114322 BGT$   114320
              BLE$   114310 BLT$   114332 BNE$   114330
              BRA$   114324 NMI$1I 114302 NMI$1M 114272
11433E 000002 $AOTS  114336
114340 000100 CCI$   114340 CDI$   114340 CFI$   114354
              $IC    114340 $ID    114340 $IR    114354
114440 000012 $WAIT  114440
114452 000026 SAL$IP 114452 SAL$MP 114470 SAL$SP 114454
              SVL$IP 114460 SVL$MP 114474 SVL$SP 114462
114500 000026 SAL$IM 114500 SAL$MM 114516 SAL$SM 114502
              SVL$IM 114506 SVL$MM 114522 SVL$SM 114510
114526 000044 JMC$   114534 JMI$M  114530 JMI$P  114526
114572 000030 AND$   114576 EQV$   114604 IOR$   114572
              XOR$   114606
114622 000014 AIF$   114622
114636 000036 CAI$   114636 CAL$   114644
114674 000016 ABS    114674
114712 000102 MOI$IA 114736 MOI$IM 114732 MOI$IS 114726
              MOI$MA 114752 MOI$MM 114746 MOI$MS 114742
              MOI$SA 114722 MOI$SM 114716 MOI$SS 114712
              MOI$0A 114766 MOI$0M 114762 MOI$0S 114756
              MOI$1A 115006 MOI$1M 115000 MOI$1S 114772
              MOL$IS 114726 MOL$SS 114712 REL$   114726
115014 000046 CMI$IP 115014 CMI$MP 115024 CMI$PI 115046
              CMI$PM 115054 CMI$PP 115034 CMI$PS 115040
              CMI$SP 115016
115062 000044 CMI$II 115102 CMI$IM 115106 CMI$IS 115076
              CMI$MI 115116 CMI$MM 115122 CMI$MS 115112
              CMI$SI 115066 CMI$SM 115072 CMI$6S 115062
115126 000034 DCI$A  115156 DCI$M  115150 DCI$P  115154
              DCI$S  115144 ICI$A  115140 ICI$M  115132
              ICI$P  115136 ICI$S  115126
115162 000046 SUI$IP 115162 SUI$MP 115176 SUI$PA 115222
              SUI$PM 115214 SUI$PP 115172 SUI$PS 115206
              SUI$SP 115164
115230 000044 SUI$IA 115250 SUI$IM 115254 SUI$IS 115244
              SUI$MA 115264 SUI$MM 115270 SUI$MS 115260
              SUI$SA 115234 SUI$SM 115240 SUI$SS 115230
115274 000046 ADI$IP 115274 ADI$MP 115310 ADI$PA 115334
              ADI$PM 115326 ADI$PP 115304 ADI$PS 115320
              ADI$SP 115276
115342 000044 ADI$IA 115362 ADI$IM 115366 ADI$IS 115356
              ADI$MA 115376 ADI$MM 115402 ADI$MS 115372
              ADI$SA 115346 ADI$SM 115352 ADI$SS 115342
115406 000064 MOI$IP 115406 MOI$MP 115422 MOI$PA 115446
              MOI$PM 115440 MOI$PP 115416 MOI$PS 115432
              MOI$SP 115410 MOI$0P 115454 MOI$1P 115462
115472 000174 CMF$II 115546 CMF$IM 115630 CMF$IP 115600
              CMF$IS 115512 CMF$MI 115536 CMF$MM 115620
              CMF$MP 115570 CMF$MS 115476 CMF$PI 115532
              CMF$PM 115614 CMF$PP 115564 CMF$PS 115472
              CMF$SI 115552 CMF$SM 115634 CMF$SP 115604
              CMF$SS 115520 $CMR   115520
```

```
        115666 000116 MOL$IA  115766 MOL$IM  115760 MOL$IP  115774
                      MOL$MA  115712 MOL$MM  115706 MOL$MP  115730
                      MOL$MS  115676 MOL$PA  115752 MOL$PM  115740
                      MOL$PP  115724 MOL$PS  115746 MOL$SA  115672
                      MOL$SM  115666 MOL$SP  115716
        116004 000026 MOF$RA  116022 MOF$RM  116012 MOF$RP  116026
                      MOF$RS  116004
        116032 000012 MOF$SS  116032
        116044 000014 MOF$IS  116044 MOF$0S  116052
        116060 000016 MOF$MS  116060 MOF$PS  116072
        116076 000014 MOF$SA  116076
        116112 000010 MOF$IA  116112
        116122 000014 MOF$SM  116122 MOF$SP  116132
        116136 000020 MOF$IM  116136 MOF$IP  116150
        116156 000020 MOF$0A  116166 MOF$0M  116156 MOF$0P  116172
        116176 000042 MOF$MA  116210 MOF$MM  116176 MOF$MP  116216
                      MOF$PA  116230 MOF$PM  116224 MOF$PP  116234
        116240 000054 SAI$PM  116240 SAI$PP  116266 SVI$PM  116252
                      SVI$PP  116300
        116314 000032 SAI$IP  116314 SAI$MP  116336 SAI$SP  116316
                      SVI$IP  116324 SVI$MP  116342 SVI$SP  116326
        116346 000032 SAI$IM  116346 SAI$MM  116370 SAI$SM  116350
                      SVI$IM  116356 SVI$MM  116374 SVI$SM  116360
        116400 000032 TSD$I   116420 TSD$M   116414 TSD$P   116424
                      TSD$S   116404 TSF$I   116420 TSF$M   116414
                      TSF$P   116424 TSF$S   116410 TSI$I   116420
                      TSI$M   116414 TSI$P   116424 TSI$S   116400
        116432 000034 CPD$SM  116454 CPF$SM  116442 CPI$SM  116436
                      CPL$SM  116432
        116466 000030 LEQ$    116470 LGE$    116500 LGT$    116476
                      LLE$    116466 LLT$    116512 LNE$    116510
        116516 000024 TSL$I   116526 TSL$M   116522 TSL$P   116534
                      TSL$S   116516
        116542 000044 SAF$IP  116542 SAF$MP  116576 SAF$SP  116544
                      SVF$IP  116554 SVF$MP  116602 SVF$SP  116556
        116606 000044 SAF$IM  116606 SAF$MM  116642 SAF$SM  116610
                      SVF$IM  116620 SVF$MM  116646 SVF$SM  116622
        116652 000036 NGD$A   116704 NGD$M   116664 NGD$P   116700
                      NGD$S   116652 NGF$A   116704 NGF$M   116664
                      NGF$P   116700 NGF$S   116652
        116710 000044 SAD$IM  116710 SAD$MM  116744 SAD$SM  116712
                      SVD$IM  116724 SVD$MM  116750 SVD$SM  116726
        116754 000030 MOD$MS  116764 MOD$PS  116760 MOD$SS  116754
                      MOD$VS  116766
LUND    117004 000054 LUNDEF  117004
CLOSU   117060 000032 CLOSEU  117060
$M.TVT  117112 000062 TVD$    117126 TVF$    117120 TVI$    117150
                      TVL$    117112 TVP$    117142 TVQ$    117134
OT      117174 001510 $ERR$$  117704 $FPERR  117546 $OTI    117174
STOP    120704 000112 EXIT    120730 FOO$    120704 STP$    120730
RIO     121016 000600 DEF$    121532 IRR$    121016 IRW$    121022
                      $GETIN  121402
GETREC  121616 000346 $GETRE  121616 $TTYIN  122116
ENDFIL  122164 000042 EOF$    122164
```

```
CLOSE    122226 000550 $CLOSE  122226
OUTREC   122776 000414 $PUTRE  122776
$M.FIO   123412 000216 $FMTDR  123412 $FMTDW  123446 $INITI  123520
OPEN     123630 000610 $OPEN   123630
FADD     124440 000062 ADF$IS  124440 ADF$MS  124452 ADF$PS  124446
                       ADF$SS  124444 SUF$IS  124466 SUF$MS  124500
                       SUF$PS  124474 SUF$SS  124472 $ADR    124444
                       $SBR    124472
FDIV     124522 000034 DIF$IS  124540 DIF$MS  124526 DIF$PS  124522
                       DIF$SS  124544 $DVR    124544
FMUL     124556 000034 MUF$IS  124574 MUF$MS  124562 MUF$PS  124556
                       MUF$SS  124600 $MLR    124600
RWBLK    124612 000460 $EOFIL  125222 $GETBL  125034 $PUTBL  124612
ERRTB    125272 000100 $ERRTB  125272
ERRS     125372 002570 $ERRS   125372
$M.LCV   130162 000106 LCI$    130162 LCO$    130230
ADDM     130270 000116 ADF$IM  130270 ADF$MM  130302 ADF$PM  130276
                       ADF$SM  130312 SUF$IM  130326 SUF$MM  130362
                       SUF$PM  130356 SUF$SM  130334
ADDP     130406 000114 ADF$IP  130406 ADF$MP  130420 ADF$PP  130414
                       ADF$SP  130430 SUF$IP  130444 SUF$MP  130500
                       SUF$PP  130474 SUF$SP  130452
ADDA     130522 000150 ADF$IA  130522 ADF$MA  130554 ADF$PA  130550
                       ADF$SA  130532 SUF$IA  130600 SUF$MA  130634
                       SUF$PA  130630 SUF$SA  130606 $FPAR   130574
                       $FPSR   130654
$M.LNO   130672 000022 COI$S   130672 COL$A   130702 COL$M   130676
                       COL$P   130706 COL$S   130672

SEGMENT PARAMETER TABLE

SEG  SIZE   LIMIT
 0  110714 130714

PROGRAM SIZE = 110714
DATA AREA SIZE = 000000
TRANSFER ADDRESS = 020004
STACK SIZE = 001000
```

```
MIDAS FORTRAN IV           11 AUG 1983  5:04:57 PM       PAGE 001

0001             SUBROUTINE HISTO(LABEL,IH,XL,XD,N)
           C     WRITTEN BY G.PHILLIPS JUNE 1981
           C
0002             REAL*8 LABEL(4)
0003             LOGICAL*1 A(50),ASTAR,ACOLON
0004             DIMENSION IH(21)
0005             DATA A/50*1H*/,IOUT/6/
0006             DATA ASTAR/1H*/,ACOLON/1H:/
           C
0007             MAX=0
0008             DO 200 K=1,N
0009     200     IF(MAX.LE.IH(K)) MAX=IH(K)
0011             IDIV=1
0012             L=5
0013     250     IF(MAX/IDIV.LE.50) GO TO 300
0015             IDIV=IDIV*2
0016             IF(IDIV.LT.L) GO TO 250
0018             IDIV=L*2
0019             L=L*10
0020             GO TO 250
0021     300     IDELT=IDIV/2
0022             WRITE(IOUT,310) LABEL
0023     310     FORMAT(1H1,4A8)
0024             WRITE(IOUT,320) IDIV
0025     320     FORMAT(1H0,'SCALE FACTOR=',I6/)
0026             X=XL
0027             IT=0
0028             DO 400 K=1,N
0029             IT=IT+IH(K)
0030             J=(IH(K)+IDELT)/IDIV
0031             IF(J.LT.1) A(1)=ACOLON
0033             WRITE(IOUT,360) X,IH(K),(A(I),I=1,J)
0034     360     FORMAT(1X,G11.3,I8,2X,50A1)
0035             A(1)=ASTAR
0036             X=X+XD
0037     400     CONTINUE
0038             WRITE(IOUT,410) IT
0039     410     FORMAT(1H0,4X,'TOTAL =',I8)
0040             DO 500 I=1,5
0041     500     WRITE(IOUT,510)
0042     510     FORMAT(1H0)
0043             RETURN
0044             END
10038             IF(RE1.LT.0.)RE1=32768.+RE1
0040             IF(RE1.GE.16384.)RE1=RE1-16384.
0042             ARRAY(K)=RE1*65536.+RE2
0043     100     CONTINUE
0044             K=M+1
0045             END FILE FILE
0046     200     M=K-1
0047             RETURN
0048             END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

LABEL    000014  REAL*8    PARAMETER ARRAY (4)
A        000026  LOGICAL*1 ARRAY (50)
IH       000016  INTEGER*2 PARAMETER ARRAY (21)
XL       000020  REAL*4    PARAMETER VARIABLE
XD       000022  REAL*4    PARAMETER VARIABLE
N        000024  INTEGER*2 PARAMETER VARIABLE
ASTAR    000112  LOGICAL*1 VARIABLE
ACOLON   000113  LOGICAL*1 VARIABLE
IOUT     000110  INTEGER*2 VARIABLE
MAX      000210  INTEGER*2 VARIABLE
K        000212  INTEGER*2 VARIABLE
IDIV     000214  INTEGER*2 VARIABLE
L        000216  INTEGER*2 VARIABLE
IDELT    000220  INTEGER*2 VARIABLE
X        000222  REAL*4    VARIABLE
IT       000226  INTEGER*2 VARIABLE
J        000230  INTEGER*2 VARIABLE
I        000232  INTEGER*2 VARIABLE
000234   REAL*4    VARIABLE
RE2      000230  REAL*4    VARIABLE

COMMON BLOCK /HEAD/     LENGTH 000142

IHDR     000000  INTEGER*2 VARIABLE
TITLE    000002  INTEGER*2 ARRAY (32)
ID       000102  INTEGER*2 ARRAY (8)
ELTIME   000122  REAL*4    VARIABLE
PLTIME   000126  REAL*4
```

```
MIDAS FORTRAN IV          11 AUG 1983  5:07:40 PM      PAGE 001

              SUBROUTINE PIVOT(A,P,E,K)
      C
      C       WRITTEN BY: G. W. PHILLIPS AND B. G. GLAGOLA JULY,1982  NRL
      C
      C       REFERENCE:  M. J. GARSIDE, APPL. STAT. 20(1971) 111-112
      C
              INTEGER P1,P2,P
              REAL A(10,10)
              LOGICAL*1 E(10)
      C    DIMENSION N MUST BE GREATER THAN OR EQUAL K.
      C    ARRAY E MUST BE INITIALIZED ".FALSE." BEFORE
      C    FIRST CALL TO PIVOT.
      C
              AA=1.0/A(P,P)
              A(P,P)=-AA
              P1=P-1
              P2=P+1
              IF(P.EQ.1) GO TO 350
              DO 300 I=1,P1
                AIP=A(I,P)*AA
                  DO 100 J=I,P1
                  A(I,J)=A(I,J)-AIP*A(J,P)
         100  CONTINUE
                  DO 200 J=P2,K
                  A(I,J)=A(I,J)-AIP*A(P,J)
         200  CONTINUE
              A(I,P)=-AIP
              IF(E(P)) A(I,P)=AIP
         300  CONTINUE
         350  DO 500 I=P2,K
              AIP=A(P,I)*AA
                  DO 400 J=I,K
                  A(I,J)=A(I,J)-AIP*A(P,J)
         400  CONTINUE
              A(P,I)=-AIP
              IF(E(P)) A(P,I)=AIP
         500  CONTINUE
              E(P)=.NOT.E(P)
              RETURN
              END
              IF(RE1.LT.0.)RE1=32768.+RE1
              IF(RE1.GE.16384.)RE1=RE1-16384.
              ARRAY(K)=RE1*65536.+RE2
       100    CONTINUE
              K=M+1
              END FILE FILE
       200    M=K-1
              RET
              END
```

```
MIDAS FORTRAN IV   STORAGE MAP


NAME     OFFSET  ATTRIBUTES

A        000014  REAL*4     PARAMETER ARRAY (10,10) VECTORED
E        000020  LOGICAL*1  PARAMETER ARRAY (10)
P        000016  INTEGER*2  PARAMETER VARIABLE
K        000022  INTEGER*2  PARAMETER VARIABLE
P1       000050  INTEGER*2  VARIABLE
P2       000052  INTEGER*2  VARIABLE
AA       000054  REAL*4     VARIABLE
I        000060  INTEGER*2  VARIABLE
AIP      000062  REAL*4     VARIABLE
J        000066  INTEGER*2  VARIABLE
```

```
MIDAS FORTRAN IV            11 AUG 1983  5:03:23 PM      PAGE 001

0001            SUBROUTINE GET(ASC,ARRAY,IVAR,FILE,M)
      C         LAST MODIFIED OCTOBER 1982 BY G.W.P.
      C
0002            COMMON/HEAD/IHDR,TITLE(32),ID(8),ELTIME,PLTIME,ERTIME,PRTIME
0003            INTEGER*2 TITLE,ID
0004            DIMENSION ARRAY(256),ITIME(4)
0005            LOGICAL*1 ASC(60),DEF(10),NAM(60)
0006            INTEGER FILE
0007            DEFINE FILE FILE(0,2,U,IVAR)
0008            ENCODE(8,10,DEF)FILE
0009  10        FORMAT('DEF ',I2,', ')
0010            DEF(9)=.FALSE.
0011            DEF(10)=.FALSE.
0012            CALL CONCAT(DEF,ASC,NAM,59)
0013            CALL LUNDEF(NAM,E)
0014            IVAR=12
0015            DO 12 I=1,4
0016  12        READ(FILE'IVAR)MLT,ITIME(I)
0017            PLTIME=ITIME(1)
0018            ELTIME=ITIME(2)
0019            PRTIME=ITIME(3)
0020            ERTIME=ITIME(4)
0021            IVAR=84
0022            DO 20 I=1,31,2
0023  20        READ(FILE'IVAR)TITLE(I),TITLE(I+1)
0024            DO 30 I=1,7,2
0025  30        READ(FILE'IVAR)ID(I),ID(I+1)
0026            IHDR=1
0027            IVAR=28
0028            READ(FILE'IVAR)A0
0029            READ(FILE'IVAR)B0
0030            READ(FILE'IVAR)C0
0031            IVAR=193
0032            DO 100 K=1,M
0033            READ(FILE'IVAR,END=200)MSB,LSB
0034            RE1=MSB
0035            RE2=LSB
0036            IF(RE2.LT.0.)RE2=65536.+RE2
0038            IF(RE1.LT.0.)RE1=32768.+RE1
0040            IF(RE1.GE.16384.)RE1=RE1-16384.
0042            ARRAY(K)=RE1*65536.+RE2
0043  100       CONTINUE
0044            K=M+1
0045            END FILE FILE
0046  200       M=K-1
0047            RETURN
0048            END
```

```
MIDAS FORTRAN IV   STORAGE MAP


NAME     OFFSET  ATTRIBUTES

ARRAY    000016  REAL*4     PARAMETER ARRAY (256)
ITIME    000026  INTEGER*2  ARRAY (4)
ASC      000014  LOGICAL*1  PARAMETER ARRAY (60)
DEF      000036  LOGICAL*1  ARRAY (10)
NAM      000050  LOGICAL*1  ARRAY (60)
IVAR     000020  INTEGER*2  PARAMETER VARIABLE
FILE     000022  INTEGER*2  PARAMETER VARIABLE
M        000024  INTEGER*2  PARAMETER VARIABLE
CONCAT   000000  REAL*4     PROCEDURE
LUNDEF   000000  INTEGER*2  PROCEDURE
E        000172  REAL*4     VARIABLE
I        000176  INTEGER*2  VARIABLE
MLT      000200  INTEGER*2  VARIABLE
A0       000202  REAL*4     VARIABLE
B0       000206  REAL*4     VARIABLE
C0       000212  REAL*4     VARIABLE
K        000216  INTEGER*2  VARIABLE
MSB      000220  INTEGER*2  VARIABLE
LSB      000222  INTEGER*2  VARIABLE
RE1      000224  REAL*4     VARIABLE
RE2      000230  REAL*4     VARIABLE

COMMON BLOCK /HEAD/      LENGTH 000142

IHDR     000000  INTEGER*2  VARIABLE
TITLE    000002  INTEGER*2  ARRAY (32)
ID       000102  INTEGER*2  ARRAY (8)
ELTIME   000122  REAL*4     VARIABLE
PLTIME   000126  REAL*4     VARIABLE
ERTIME   000132  REAL*4     VARIABLE
PRTIME   000136  REAL*4     VARIABLE
```

```
MIDAS FORTRAN IV          11 AUG 1983  5:08:44 PM       PAGE 001

0001            SUBROUTINE PUT(ASC,ARRAY,IVAR,FILE,M)
        C       WRITTEN BY G. PHILLIPS, JUNE 1981
        C
0002            DIMENSION ARRAY(256)
0003            LOGICAL*1 ASC(60),DEF(10),NAM(60)
0004            INTEGER FILE
0005            DEFINE FILE FILE(448,2,U,IVAR)
0006            ENCODE(8,10,DEF)FILE
0007     10     FORMAT('DEF ',I2,', ')
0008            DEF(9)=.FALSE.
0009            DEF(10)=.FALSE.
0010            CALL CONCAT(DEF,ASC,NAM,59)
0011            CALL LUNDEF(NAM,E)
0012            IVAR=193
0013            DO 100 K=1,M
0014            V=ARRAY(K)
0015            IF(V.LT.0.) V=0.
0017            IF(V.LT.65536.) GOTO 50
0019            V=V/65536.
0020            IF(V.GT.16383.) V=16383.
0022            MSB=V
0023            LSB=0
0024            GOTO 80
0025      50    IF(V.GT.32767.) V=V-65536.
0027            LSB=V
0028            MSB=0
0029      80    WRITE(FILE'IVAR)MSB,LSB
0030      100   CONTINUE
0031            END FILE FILE
0032            CALL CLOSEU(FILE)
0033            RETURN
0034            END
A0038            IF(RE1.LT.0.)RE1=32768.+RE1
0040            IF(RE1.GE.16384:.)RE1=RE1-16384.
0042            ARRAY(K)=RE1*65536.+RE2
0043     100    CONTINUE
0044            K=M+1
0045            END FILE FILE
0046     200    M=K-1
0047            RETURN
0048            END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

ARRAY    000016  REAL*4    PARAMETER ARRAY (256)
ASC      000014  LOGICAL*1 PARAMETER ARRAY (60)
DEF      000026  LOGICAL*1 ARRAY (10)
NAM      000040  LOGICAL*1 ARRAY (60)
IVAR     000020  INTEGER*2 PARAMETER VARIABLE
FILE     000022  INTEGER*2 PARAMETER VARIABLE
M        000024  INTEGER*2 PARAMETER VARIABLE
CONCAT   000000  REAL*4    PROCEDURE
LUNDEF   000000  INTEGER*2 PROCEDURE
E        000172  REAL*4    VARIABLE
K        000176  INTEGER*2 VARIABLE
V        000200  REAL*4    VARIABLE
MSB      000204  INTEGER*2 VARIABLE
LSB      000206  INTEGER*2 VARIABLE
CLOSEU   000000  REAL*4    PROCEDURE
000050   INTEGER*2 VARIABLE
P2       000052  INTEGER*2 VARIABLE
AA       000054  REAL*4    VARIABLE
I        000060  INTEGER*2 VARIABLE
AIP      000062  REAL*4    VARIABLE
J        000066  INTEGER*2 VARIABLE
```

```
MIDAS FORTRAN IV          11 AUG 1983  5:10:02 PM      PAGE 001

      SUBROUTINE RESID(R,NZ,MF)
C     WRITTEN BY G.PHILLIPS JUNE 1981
C
      DIMENSION R(256),A(4),C(4),IH(21)
      REAL*8 LABEL(4)
      DATA IOUT/6/
      DATA LABEL/8H    DIST,8HRIBUTION,8H OF RESI,8HDUALS   /
C
      DO 100 K=1,21
  100 IH(K)=0
      DO 150 J=1,4
  150 C(J)=0.
      DO 200 I=NZ,MF
      DO 180 J=1,4
  180 C(J)=C(J)+R(I)**J
      K=R(I)*2.+11.5
      IF(K.GT.21) K=21
      IF(K.LT.1) K=1
      IH(K)=IH(K)+1
  200 CONTINUE
      M=MF-NZ+1
      DO 220 J=1,4
  220 C(J)=C(J)/M
      U=C(1)
      A(1)=U
      U2=U*U
      U3=U*U2
      U4=U*U3
      V=C(2)-U2
      A(2)=V
      A(3)=C(3)-3*U*C(2)+2*U3
      A(4)=C(4)-4*U*C(3)+6*U2*C(2)-3*U4
      SD=SQRT(V)
      SQ=A(3)/SD**3
      EX=A(4)/V**2-3.
D     WRITE(IOUT,1000)M,(I,A(I),C(I),I=1,4)
D1000 FORMAT(1H0,'NUMBER OF ELEMENTS =',I4/
D    1                      (I5,2G11.3))
      WRITE(IOUT,300)U,V,SQ,EX
  300 FORMAT(1H0,'MEAN RESIDUAL =',F8.3,', VARIANCE =',F8.3,
     1    ', SKEWNESS =',G11.3,', EXCESS =',G11.3)
      XL=-5.0
      XD=0.5
      N=21
      CALL HISTO(LABEL,IH,XL,XD,N)
      RETURN
      END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

R        000014  REAL*4     PARAMETER ARRAY (256)
A        000022  REAL*4     ARRAY (4)
C        000042  REAL*4     ARRAY (4)
IH       000062  INTEGER*2  ARRAY (21)
LABEL    000134  REAL*8     ARRAY (4)
NZ       000016  INTEGER*2  PARAMETER VARIABLE
MF       000020  INTEGER*2  PARAMETER VARIABLE
IOUT     000174  INTEGER*2  VARIABLE
K        000306  INTEGER*2  VARIABLE
J        000310  INTEGER*2  VARIABLE
I        000312  INTEGER*2  VARIABLE
M        000314  INTEGER*2  VARIABLE
U        000316  REAL*4     VARIABLE
U2       000322  REAL*4     VARIABLE
U3       000326  REAL*4     VARIABLE
U4       000332  REAL*4     VARIABLE
V        000336  REAL*4     VARIABLE
SD       000342  REAL*4     VARIABLE
SQRT     000000  REAL*4     PROCEDURE
SQ       000346  REAL*4     VARIABLE
EX       000352  REAL*4     VARIABLE
XL       000356  REAL*4     VARIABLE
XD       000362  REAL*4     VARIABLE
N        000366  INTEGER*2  VARIABLE
HISTO    000000  REAL*4     PROCEDURE
```

```
MIDAS FORTRAN IV           11 AUG 1983  5:11:14 PM      PAGE 001

      SUBROUTINE RSTAT(NCH,P,R,NZ,MF)
C     WRITTEN BY T. B. GOSNELL 22 JUNE, 1982    LLNL
C     MODIFIED FOR PREGA BY B. G. GLAGOLA  20 JULY, 1982  NRL
C
C     THIS PROGRAM ANALYSES RESIDUALS FROM PREGA, COMPUTES A
C     A RESIDUAL STATISTIC, ITS EXPECTATION VALUE, STANDARD
C     DEVIATION AND AN APPROXIMATE ASSOCIATED FAP.
C
C     REFERENCE:  A. AARNIO, M. J. KOSKELO AND P. ZOMBORI,
C                 NUCLEAR INSTR. AND METH. 184(487)1981.
C
      DIMENSION R(512)
      REAL NRMR
      INTEGER P
      CH=NCH
      EP=P
      DF=CH-EP
      MF1=MF-1
      RSTT=0
      DO 10 I=NZ,MF1
      RSTT=RSTT+R(I)*R(I+1)
   10 CONTINUE
      RSTT=RSTT+R(1)*R(MF)
      RSTT=RSTT/DF
      EXRES=-EP/CH
      SIGRES=SQRT(CH)/DF
      NRMR=(RSTT-EXRES)/SIGRES
      CALL XQCALC(DF,0.,1.,NRMR,Q,1)
      WRITE(6,50)
   50 FORMAT(' ',////' ***** LLNL R-STATISTIC  *****')
      WRITE(6,51) RSTT,EXRES,SIGRES,NRMR,Q
   51 FORMAT('0','AUTO-CORRELATION COEFFICIENT',17('.'),1PE12.4/
     1' EXPECTATION VALUE',28('.'),1PE12.4/
     2' STANDARD DEVIATION',27('.'),1PE12.4/
     3' EQUIVALENT NORMAL STANDARD DEVIATIONS',8('.'),1PE12.4/
     4' R-STAT FAP',35('.'),1PE12.4)
      RETURN
      END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME    OFFSET  ATTRIBUTES

R       000020  REAL*4    PARAMETER ARRAY (512)
NCH     000014  INTEGER*2 PARAMETER VARIABLE
P       000016  INTEGER*2 PARAMETER VARIABLE
NZ      000022  INTEGER*2 PARAMETER VARIABLE
MF      000024  INTEGER*2 PARAMETER VARIABLE
NRMR    000404  REAL*4    VARIABLE
CH      000410  REAL*4    VARIABLE
EP      000414  REAL*4    VARIABLE
DF      000420  REAL*4    VARIABLE
MF1     000424  INTEGER*2 VARIABLE
RSTT    000426  REAL*4    VARIABLE
I       000432  INTEGER*2 VARIABLE
EXRES   000434  REAL*4    VARIABLE
SIGRES  000440  REAL*4    VARIABLE
SQRT    000000  REAL*4    PROCEDURE
XQCALC  000000  REAL*4    PROCEDURE
Q       000444  REAL*4    VARIABLE
LE
EX      000352  REAL*4    VARIABLE
XL      000356  REAL*4    VARIABLE
XD      000362  REAL*4    VARIABLE
N       000366  INTEGER*2 VARIABLE
HISTO   000000  REAL*4    PROCEDURE
```

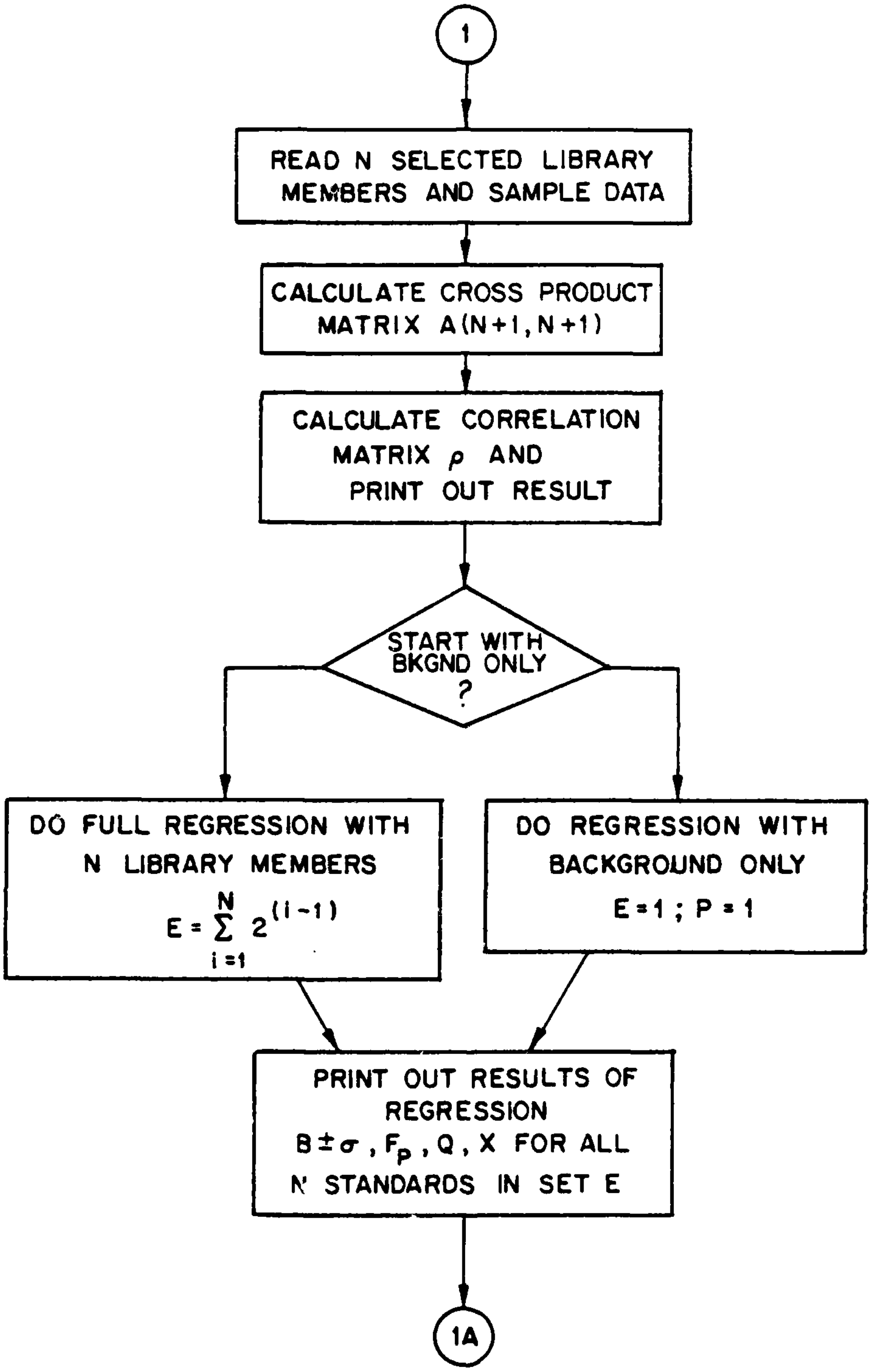
```
MIDAS FORTRAN IV          11 AUG 1983  5:12:05 PM      PAGE 001

      SUBROUTINE RUNS(R,NZ,MF)
C     WRITEN BY G.PHILLIPS JUNE 1981
C
      DIMENSION R(256),IH(21),NCH(50),KSI(12),NRS(50)
      REAL*8 LABEL(4)
      DATA LABEL/8H      DI,8HSTRIBUTI,8HON OF RU,8HNS      /
      DATA KSI/-10,-9,-8,-7,-6,-5,5,6,7,8,9,10/
C
      DO 100 K=1,21
  100 IH(K)=0
      I=NZ
      J=0
C
  200 K=0
      ICH=I
  210 IF(R(I).LT.0.) GOTO 300
      K=K+1
      I=I+1
      IF(I.GT.MF) GOTO 400
      IF(R(I).GE.0.) GOTO 210
      GOTO 400
 300  K=K-1
      I=I+1
      IF(I.GT.MF) GOTO 400
      IF(R(I).LE.0.) GOTO 300
C
  400 K=K+11
      IF(K.LT.1) K=1
      IF(K.GT.21) K=21
      IH(K)=IH(K)+1
      IF(K.GT.6.AND.K.LT.16) GO TO 450
      IF(K.GT.6) K=K-9
      J=J+1
      NCH(J)=ICH
      NRS(J)=KSI(K)
   450 IF(I.LE.MF) GO TO 200
C
      XL=-10.0
      XD=1.0
      N=21
      CALL HISTO(LABEL,IH,XL,XD,N)
      WRITE(6,500)
   500 FORMAT(' ','LOCATIONS AND SIZES OF LARGE RUNS'//)
      WRITE(6,501) (NCH(I),NRS(I),I=1,J)
   501 FORMAT(' ',5(5X,I4,I6))
      RETURN
      END
```

```
MIDAS FORTRAN IV   STORAGE MAP

NAME    OFFSET  ATTRIBUTES

R       000014  REAL*4    PARAMETER ARRAY (256)
IH      000022  INTEGER*2 ARRAY (21)
NCH     000074  INTEGER*2 ARRAY (50)
KSI     000240  INTEGER*2 ARRAY (12)
NRS     000270  INTEGER*2 ARRAY (50)
LABEL   000434  REAL*8    ARRAY (4)
NZ      000016  INTEGER*2 PARAMETER VARIABLE
MF      000020  INTEGER*2 PARAMETER VARIABLE
K       000564  INTEGER*2 VARIABLE
I       000566  INTEGER*2 VARIABLE
J       000570  INTEGER*2 VARIABLE
ICH     000572  INTEGER*2 VARIABLE
XL      000574  REAL*4    VARIABLE
XD      000600  REAL*4    VARIABLE
N       000604  INTEGER*2 VARIABLE
HISTO   000000  REAL*4    PROCEDURE
LE
EX      000352  REAL*4    VARIABLE
XL      000356  REAL*4    VARIABLE
XD      000362  REAL*4    VARIABLE
N       000366  INTEGER*2 VARIABLE
HISTO   000000  REAL*4    PROCEDURE
```

```
MIDAS FORTRAN IV          11 AUG 1983  5:13:16 PM       PAGE 001

      SUBROUTINE XQCALC(DF,F,ZS,X,Q,IEN)
C
C     WRITTEN BY B. G. GLAGOLA  JUNE, 1982   NRL
C
C
C
C   THIS ROUTINE CALCULATES X(F) AND Q(X) GIVEN F
C
      IF(IEN.EQ.1) GO TO 40
      DF2=2.*DF-1
      IF(F.GT.DF2) GO TO 50
      X=(SQRT(F)*(1.-1./(4.*DF)))/(SQRT(1.-F/(2.*DF)))
   40 IF(X.GT.12.9) GO TO 60
      Z=0.398942*EXP(-X**2/2.)
      T=1.0/(1.0+0.33267*X)
      Q=Z*(.436184*T-.120168*T**2+.937298*T**3)
      GO TO 70
   50 X=9.9999E09
   60 Q=0.00
   70 IF(ZS.LT.0.0) Q=1.-Q
      IF(ZS.LT.0.0)X=-X
      RETURN
      END
 400  K=K+11
      IF(K.LT.1) K=1
      IF(K.GT.21) K=21
      IH(K)=IH(K)+1
      IF(K.GT.6.AND.K.LT.16) GO TO 450
      IF(K.GT.6) K=K-9
      J=J+1
      NCH(J)=ICH
      NRS(J)=KSI(K)
  450 IF,J)
  501 FORMAT(' ',5(5X,I4,I6))
      RETURN
      END
```

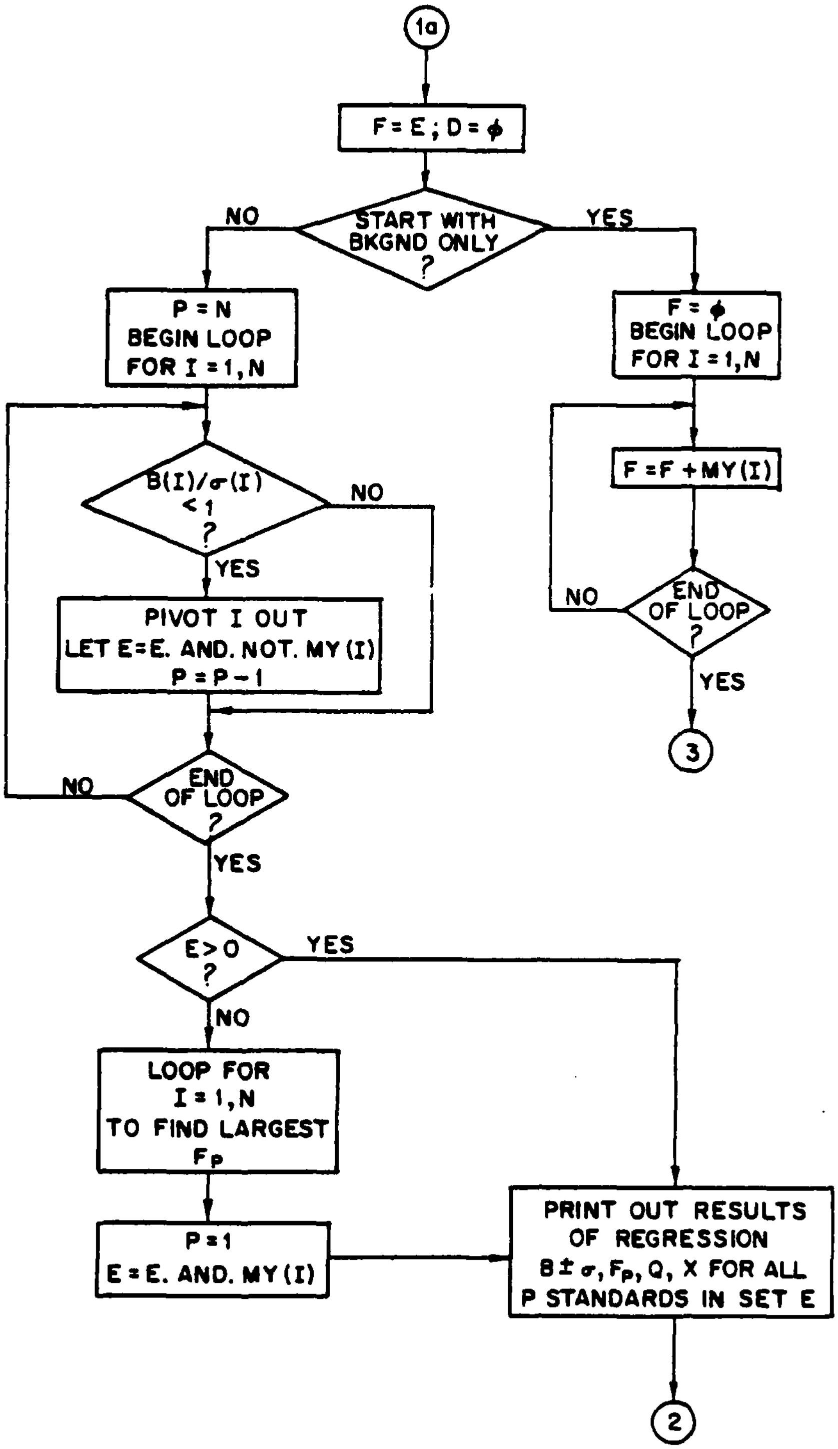
```
MIDAS FORTRAN IV   STORAGE MAP

NAME     OFFSET  ATTRIBUTES

DF       000014  REAL*4     PARAMETER VARIABLE
F        000016  REAL*4     PARAMETER VARIABLE
ZS       000020  REAL*4     PARAMETER VARIABLE
X        000022  REAL*4     PARAMETER VARIABLE
Q        000024  REAL*4     PARAMETER VARIABLE
IEN      000026  INTEGER*2  PARAMETER VARIABLE
DF2      000064  REAL*4     VARIABLE
SQRT     000000  REAL*4     PROCEDURE
Z        000070  REAL*4     VARIABLE
EXP      000000  REAL*4     PROCEDURE
T        000074  REAL*4     VARIABLE
```

PREGA SETUP

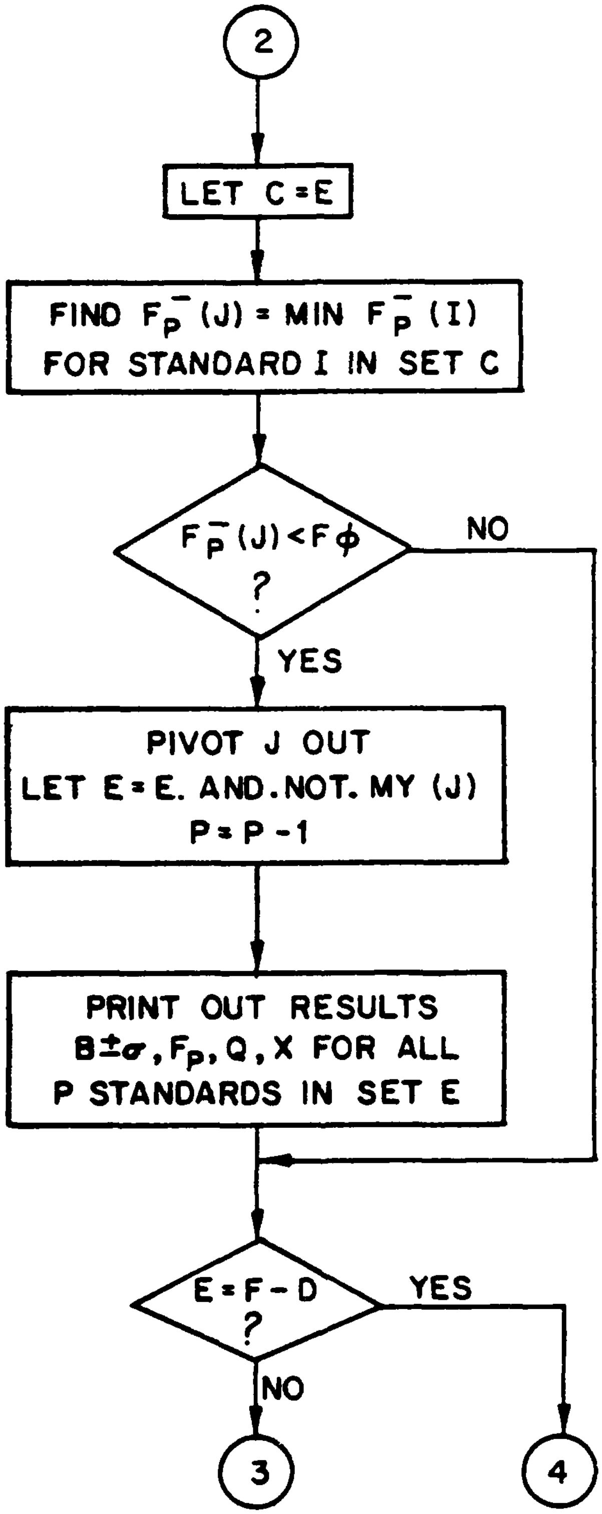

1
READ N SELECTED LIBRARY MEMBERS AND SAMPLE DATA
CALCULATE CROSS PRODUCT MATRIX A(N+1,N+1)
CALCULATE CORRELATION MATRIX ρ AND PRINT OUT RESULT
START WITH BKGND ONLY ?
DO FULL REGRESSION WITH N LIBRARY MEMBERS $E=\sum_{i=1}^{N} 2^{(i-1)}$
DO REGRESSION WITH BACKGROUND ONLY E=1 ; P=1
PRINT OUT RESULTS OF REGRESSION $B\pm\sigma$, F_p, Q, X FOR ALL N' STANDARDS IN SET E
1A

Flow Chart for the Program PREGA showing details of the Major Sections Given in Figure 1.

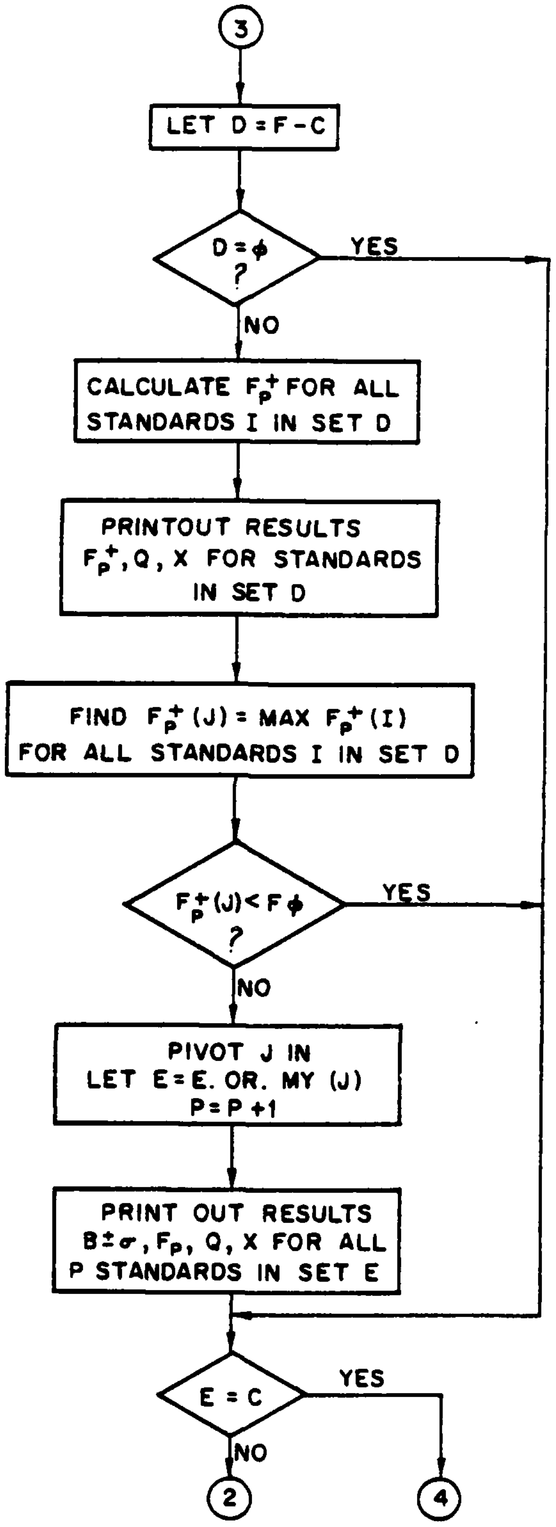
SELECT TRIAL SUBSET

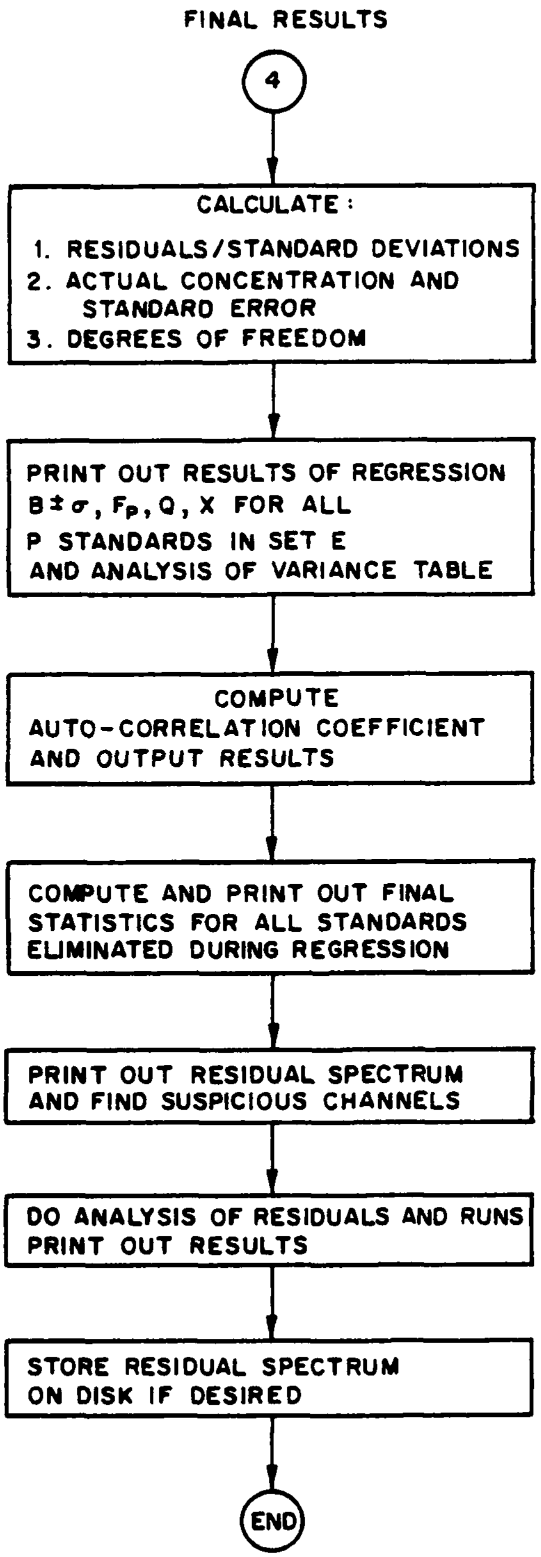


BACKWARD PIVOT



FORWARD PIVOT


3
LET D = F − C
D = φ ?
YES
NO
CALCULATE F_p^+ FOR ALL STANDARDS I IN SET D
PRINTOUT RESULTS F_p^+, Q, X FOR STANDARDS IN SET D
FIND $F_p^+(J)$ = MAX $F_p^+(I)$ FOR ALL STANDARDS I IN SET D
$F_p^+(J) < F\phi$?
YES
NO
PIVOT J IN
LET E = E. OR. MY (J)
P = P + 1
PRINT OUT RESULTS
B ± σ, F_P, Q, X FOR ALL P STANDARDS IN SET E
E = C
YES
NO
2
4


FINAL RESULTS
4
CALCULATE :
1. RESIDUALS/STANDARD DEVIATIONS
2. ACTUAL CONCENTRATION AND STANDARD ERROR
3. DEGREES OF FREEDOM
PRINT OUT RESULTS OF REGRESSION
$B \pm \sigma, F_p, Q, X$ FOR ALL
P STANDARDS IN SET E
AND ANALYSIS OF VARIANCE TABLE
COMPUTE
AUTO-CORRELATION COEFFICIENT
AND OUTPUT RESULTS
COMPUTE AND PRINT OUT FINAL
STATISTICS FOR ALL STANDARDS
ELIMINATED DURING REGRESSION
PRINT OUT RESIDUAL SPECTRUM
AND FIND SUSPICIOUS CHANNELS
DO ANALYSIS OF RESIDUALS AND RUNS
PRINT OUT RESULTS
STORE RESIDUAL SPECTRUM
ON DISK IF DESIRED
END

Appendix C: MIDAS System Subroutines

1. ATIME
2. MTAPEF
3. CLOSEU
4. CONCAT
5. LUNDEF

ATIME (ASCII Date/Time) Subroutine

The ATIME subroutine is used to return the ASCII date and time to the FORTRAN user. This subroutine is written in assembler.

Form

CALL ATIME (string)

String is the name of the 24 byte string to receive the ASCII date/time. The returned string is 24 bytes in length in the following format:

```
dd-mmm-yyyy hh:mm:ss  xx
```

where:

dd	Day (number)
mmm	First three letters of the month
yyyy	Year (number)
hh	Hour of the day (12 hours)
mm	Minutes after the hour
ss	Seconds after the minute
xx	AM or PM

NOTE: If the parameter count is <1 or >1, then a return is made to MIDAS.

Example

```
LOGICAL*1 STRING(24)
.
.
.
CALL ATIME(STRING)
.
.
.
```

MTAPEF (FORMATTED MAGNETIC TAPE) SUBROUTINE

The MTAPEF subroutine controls the magnetic tape and its related functions. The subroutine is compatible with either 7 or 9 track magnetic tape. This subroutine is written in Assembler.

Form

```
CALL MTAPEF (a,b,c,d,e)
```

where:

a = Command: INTEGER*2 variable (required argument).
= 1 - Initialize control formatter (a,b).
= 2 - Transport off-line (a,b).
= 3 - Rewind (a,b).
= 4 - Search for logical EOT (a,b).
= 5 - Search for file (a,b,c).
= 6 - Search for record (a,b,c).
= 7 - Read one record (a,b,c,d,e).
= 8 - Verify one record (a,b,c,d,e).
= 9 - Write one record (a,b,c,d).
= 10 - Not used.
= 11 - Over write one record (a,b,c,d).
= 12 - Dump one record (a,b,c,d).
= 13 - Write one filemark (a,b).
= 14 - Write a logical EOT (a,b).
= 15 - Open transport (a,b,c).
= 16 - Close transport (a,b).
= 17 - Tagword (a,b,c,d)

b = Error Number: INTEGER*2 variable (required).
= 1 - No error.
= 2 - Transport assigned to other/or no user.
= 3 - Magnetic tape transport number error.
= 4 - Segment is read only.
= 5 - Segment is not accessible for I/O.
= 6 - Memory is not contiguous.
= 7 - Cross segments have different status.
= 8 - No filemark detected for last operation.
= 9 - Filemark detected during last operation.
= 10 - Located on or past physical EOT.
= 11 - Record read less than list word #14.
= 12 - Record read greater than list word #14.
= 13 - Invalid or undefined OP code.
= 14 - Data late.
= 15 - Invalid password.
= 16 - Motion error.
= 17 - Verification error.
= 18 - Write protect error.
= 19 - Parity CRC or LRC error during read.
= 20 - Operation attempt on off-line transport.
= 21 - No logical EOT detected during operation.

= 22 - Logical EOT detected during operation.
= 23 - Magnetic tape not off-line.
= 24 - Undefined error bit in status.
= 25 - Executive error during operation.
= 26 - Illegal number of arguments.
= 27 - Illegal command number.
= 28 - Record length greater than 513 bytes for 7 track transport.

c = Command Parameter 1: INTEGER*2 variable.

Command Number	Parameter
5	# files to skip
6	# records to skip
7	# bytes to skip
8	# bytes to skip
9	# bytes to write
11	# bytes to write
12	# bytes to write
15	# transport to open

17	Subcommand code (c)
	c = 1 - Get tagword
	c = 2 - Increment tagword
	c = 3 - Put tagword

d = Command Parameter 2: INTEGER*2 variable or array name.

Command Number	Parameter
7	# bytes to read
8	# bytes to verify
9	Array name to write
11	Array name to write
12	Array name to write
17	Get, Increment, or Put tagword

e = Command parameter 3: Integer variable.

Command Number	Parameter
7	Array name to read
8	Array name to verify

The following is a test program for MTAPEF. The program does the following things:

1. Opens transport.

2. Initializes magnetic tape controller.

3. Reads from the disk and writes to the tape 4Ø96 channels in groups of 128 channels.

4. Writes logical end of tape (of double filemark).

5. Rewinds tape.

6. Reads from tape and writes to disk 4Ø96 channels in groups of 128 channels.

7. Rewinds tape.

8. Closes transport.

```
C         PROGRAM TO TEST MAG TAPE OPERATIONS
C         USING FORTRAN EXTENSION LIBRARY SUBROUTINE
C
          INTEGER ARRAY (513) A,B,C,D
          DEFINE FILE 12(4Ø96,2,U,IVAR)
C
C         OPEN FILE
C
          A=15
          C=1
          CALL MTAPEF(A,B,C)
          ID(B.NE.1)WRITE(6,11Ø)B
C
C         INITIALIZE MAG TAPE
C
          A=1
          CALL MTAPEF(A,B)
          IF(B.NE.1)WRITE(6,1ØØ)B
C
C         NOW READ THE DISK & WRITE TO MAG TAPE
C
          A=9
          C=513
          IVAR=129
          DO 2ØØ I=1,16
          DO 3ØØ J=1,512,2
3ØØ       READ(12'IVAR)ARRAY(J),ARRAY(J+1)
          CALL MTAPEF(A,B,C,ARRAY)
          IF(B.NE.1)WRITE(6,12Ø)B
2ØØ       CONTINUE
C
C         WRITE A DOUBLE FILEMARK
C
          A=14
          CALL MTAPEF(A,B)
          IF(B.NE.1)WRITE(6,13Ø)B
C
C         REWIND TAPE
C
          A=3
          CALL MTAPEF(A,B)
          IF (B.NE.1)WRITE(6,14Ø)B
```

```
C
C     NOW READ FROM TAPE & WRITE TO DISK
C
      A=7
      C=Ø
      D=513
      IVAR=129
      DO 4ØØ I=1,16
      CALL MTAPEF(A,B,C,D,ARRAY)
      IF(B.NE.1)WRITE(6,15Ø)B
      DO 5ØØ J=1,512,2
5ØØ   READ(12'IVAR)ARRAY(J),ARRAY(J+1)
4ØØ   CONTINUE
C
C     REWIND TAPE
C
      A=3
      CALL MTAPEF(A,B)
      IF(B.NE.1)WRITE(6,16Ø)B
C
C     DO A CLOSE
C
      A=16
      CALL MTAPEF(A,B)
      IF(B.NE.1)WRITE(6,17Ø)B
C
C
1ØØ   FORMAT(' ','ERROR ',I3,' IN MAG TAPE INITIALIZE')
11Ø   FORMAT(' ','ERROR ',I3,' IN MAG TAPE OPEN')
12Ø   FORMAT(' ','ERROR ',I3,' IN MAG TAPE WRITE')
13Ø   FORMAT(' ','ERROR ',I3,' IN SETTING FILE MARK')
14Ø   FORMAT(' ','ERROR ',I3,' IN FIRST REWIND')
15Ø   FORMAT(' ','ERROR ',I3,' IN MAG TAPE READ')
16Ø   FORMAT(' ','ERROR ',I3,' IN SECOND MAG TAPE REWIND')
17Ø   FORMAT(' ','ERROR ',I3,' ERROR IN MAG TAPE CLOSE')
      END
```

CLOSEU (Close with Update) Subroutine

This subroutine closes a file with update and moves the current end sector to the current end of the file. This subroutine is written in assembler.

Form

CALL CLOSEU(lun)

where:

lun = Logical unit number associated with the file.

CONCAT CONCAT

The CONCAT subroutine is used to concatenate character strings.

Form: CALL CONCAT (a, b, out(, len(, err)))

Where:

a is the array containing the left string

b is the array containing the right string.

out is the array into which the concatenated result is placed. This array must be at least one element longer than the maximum length of the result string (i. e., one greater than the value of len, if specified).

len is the integer number of characters representing the maximum length of the output string. The effect of len is to truncate the output string to a given length, if necessary.

err is the Logical error flag set if the output string is truncated to the length specified by len.

The string in array "a" immediately followed on the right by the string in array "b" and a terminating null character replaces the string in array "out". Any combination of string arguments is allowed so long as "b" and "out" do not specify the same array. Concatenation stops either when a null character is detected in "b" or when the number of characters specified by "len" have been moved.

If whether the left or right string is a null string, the other string is copied to "out". If both are null strings, then "out" is set to a null string. The old contents of "out" are lost when this routine is called.

Errors:

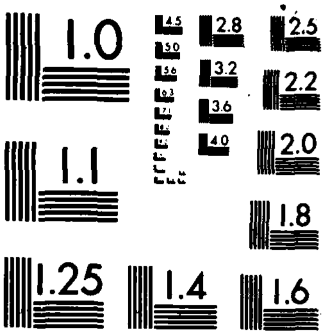
Error conditions are indicated by "err", if specified. If "err" is given and the output string would have been longer than "len" characters, then "err" is set to . TRUE. ; otherwise, "err" is unchanged.

Example:

The following example concatenates the string in array STR and the string in array IN and stores the resultant string in array OUT. OUT cannot be larger than 29 characters.

```
LOGICAL*1 IN(30), OUT(30), STR(7)
.
.
.
CALL CONCAT(STR, IN, OUT, 29)
```

AD-A141 539 PROGRAM PREGA PIVOTAL REGRESSION ANALYSIS OF GAMMA-RAY 3/3
SPECTRA FROM NAI(T..(U) NAVAL RESEARCH LAB WASHINGTON
DC G W PHILLIPS ET AL. 06 APR 84 NRL-MR-5275
UNCLASSIFIED SBI-AD-E000 564 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

```
;       LUNDEF    MOD 44-0437-02    25 AUG 77    L.HOLMES
;
;
;       COPYRIGHT (C) 1976 BY NUCLEAR DATA INCORPORATED
;       SCHAUMBURG, ILLINOIS 60196
;
;
;       THIS DOCUMENTATION IS FURNISHED INCLUDING THE COPYRIGHT
;       NUCLEAR DATA INCORPORATED. IT MUST NOT BE REMOVED AND MAY NOT
;       BE REPRODUCED OR REDISTRIBUTED WITHOUT THE WRITTEN EXPLICIT
;       PERMISSION OF NUCLEAR DATA.
;
;       WHEREBY NUCLEAR DATA CORPORATION ASSUMES THE SUPPORT
;       RESPONSIBILITIES WHEN PROPERLY APPLIED TO ND6600 (MIDAS)
;       SYSTEMS.
;        NUCLEAR DATA CORPORATION ASSUMES NO RESPONSIBILITY FOR USE
;       OF AFORMENTIONED SOFTWARE IN EQUIPMENT EXCLUDED FROM THE
;       ND6600-SERIES EQUIPMENT SUPPLIED BY NUCLEAR DATA INCORPORATED.
;
        .TITLE LUNDEF
        .CSECT LUND
        .MCALL EXEC
        .MCALL RESTORE,SAVE,.REGDEF
        .GLOBL  LUNDEF
        .REGDEF
LUNDEF: SAVE R0,R1,R2,R3,R4,R5
        TST  (R5)+
        MOV  (R5)+,R2
        EXEC CLI$,R2
        BCC  RET
        MOV  #1,@(R5)
        CLC
RET:    RESTORE R5,R4,R3,R2,R1,R0
        RTS  PC
        .END
```

END

FILMED

DTIC